.EXE

JUNE 1991 VOL 6 ISSUE 1

EXCLUSIVE PREVIEW: MICROSOFT'S VISUAL BASIC

The Software Developers' Magazine

A bumper issue celebrates Software Tools '91 - Hope to see you at the show.

We concentrate on Software Engineering: Two features pick up the CASE.

Computer charlatans beware! Blood will be shed in Jules May's new column.

AT&T's Brian Kernighan graces our UNIX pages.

Low-level technical treatment of maths coprocessors and DOS disk formats.

THE WINDOWS LEADERS

The Leading Light in Windows Training

QA Training has long been recognised as the leader in Windows Training. The world's first company to offer Windows programming courses. The first authorised by Microsoft. And the very first to give technical training in Windows 3.

Our unique range of courses – Windows training for developers, support personnel and users – are widely acknowledged to be, quite simply, the best.

We also have worldwide involvement in consulting on many of the major Windows projects with corporate developers and software suppliers.

Introduction to Windows

Use Windows mouse and keyboard controls; Customise the Windows desktop; Work with Windows file and print utilities; Use Windows applications productively; Understand the theory of Windows. *2 Days.*

Supporting Windows

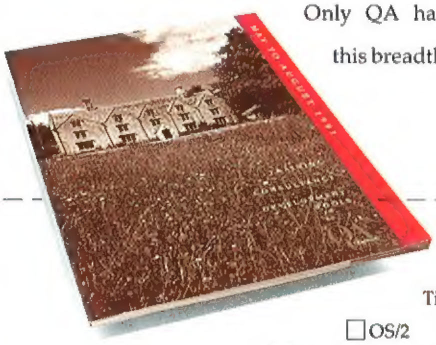
Install Windows and explore related issues; Configure Windows and applications; Use network, communications and print facilities; Understand Windows internals – memory, multi-tasking. *3 Days.*

Microsoft Windows Programming

Use the MS Windows SDK and tools; Write well structured compatible Windows applications; Understand the Windows API and Windows programming principles. *5 Days.*

Advanced Windows Programming

Understand application structure and design techniques in a non-pre-emptive multi-tasking environment; Understand the more complex aspects of MS Windows programming. Write MS Windows using advanced functions and techniques. *5 Days.*



Only QA has this breadth of experience in every aspect of the Windows environment. Experience which is your assurance that a QA Windows course, is certain to maximise your skills and effectiveness.

For more information, please complete and return the coupon today or call Samantha Trinder on 0285 655888.

Mail to QA Training Ltd, Cecily Hill Castle, Cirencester, Gloucestershire GL7 2EF. UK.

Tick here for full details:

□OS/2 □Unix □Windows
□C □Object-Oriented □Networks
□Consultancy □Products

NAME TITLE

COMPANY

ADDRESS

TEL. NO.

.EXE 6/91

QA TRAINING

The Leading Light in Windows Training

CIRCLE NO. 841

.EXE

Vol 6 Issue 1 *June 1991*

Editor:	Willie Watts
News Editor:	Dan O'Brien
Editorial Assistant:	Wendie Leek
Design & Layout:	Katerina Adams Mark English
Group Advertising Manager	Sandra Inniss-Palmer
Senior Advertising Executive:	Ed Butcher
Advertising Executive:	Chris Haill
Reader Services & Promotions:	Helena Adams
Publisher:	Jon Howell
Administration & Subscriptions:	Rena Gibbs Sandy McDonnell
Front Cover Illustration:	Chris Duggan

General

.EXE Magazine is independent and not affiliated to any vendor of hardware, software or services. It is published by Process Communications Ltd, 10 Barley Mow Passage, Chiswick, London W4 4PH.

Tel: (Advertising/Editorial/Production) 081 994 6477
(Subscriptions) 044282 4501
Facsimile: 081 994 1533 Telex: 8811418 SPACES G
ISSN: 0268-6872

Subscriptions

.EXE Magazine is a monthly journal for software developers. It is available only by subscription, at a cost of £35 per annum (11 issues) in the UK (see insert card for academic and overseas rate). 'A subscription implies that this journal will be sent to the subscriber until one of the three expires' - AG Macdonell. The magazine is published around the 26th of the month preceding the cover date. There is no January issue. To subscribe, please call 044282 4501, or write to The Subscriptions Manager, .EXE Magazine, 10 Barley Mow Passage, Chiswick, London W4 4PH. We can invoice your company or take ACCESS and VISA cards. In case of a query about your subscription, please call 044282 4501.

Back issues are available at a cost of £3.50 each - call our Chiswick office (081 994 6477) for a list of issues still in print. We also offer a range of magazine binders, disk holders etc - please call for details.

Editorial

Editorial enquiries should be addressed to The Editor, .EXE Magazine, 10 Barley Mow Passage, Chiswick, London W4 4PH. We welcome letters, opinions, suggestions and articles from our readers. If you are interested in contributing articles, please write to this office for a copy of our Contributors' Guide.

Information contained in .EXE is believed to be correct. If errors are found, we will endeavour to publish a clarification in the next available issue.

From time to time, we offer to copy the PC program code described in an article onto diskette for our readers. In this case, please send a blank, formatted disk with a self-addressed, prepaid mailer to the editorial address given above. We can copy both 5.25" and 3.5" disks.

The publishers can accept no liability for any consequences of using software distributed in this way.

Advertising

If you are interested in advertising in this magazine, please write to the Display Advertising Manager/Recruitment Sales Manager (as appropriate), .EXE Magazine, 10 Barley Mow Passage, Chiswick, London W4 4PH, or call 081 994 6477 for details of our advertising rates.

Pronunciation

The name of .EXE Magazine is pronounced to rhyme with 'not sexy magazine'.

Copyright

Issue theme: Software Engineering

Have mainframe, am stuffed

Richard Samworth has a message for the mainframe-package worshipping DP departments. He could have delivered it with two fingers of one hand, but has instead climbed on the .EXE Soapbox.

The 1980s, as we all know, was a decade of unprecedented business expansion made possible by the increased availability of capital and the rapid introduction of computer systems. In the new decade, as the dust begins to settle, what's that strange wailing noise coming from the DP department? It's all to do with a bloated mainframe project and the story goes something like this.

'Go for it' was the slogan on everyone's lips, 'We need a bigger computer. Our old systems just can't cope with all this new business'.

Management are faced with some tough decisions: rewrite the existing systems or buy packaged software? They toss a coin and it lands package side up. Salesmen appear, give presentations, show slides, arrange jollies for senior personnel and a package emerges as the winner. It was written in the New World but, no matter, increasing market globalisation and anyway, it is certainly comprehensive and benefits from one significant feature that makes it stand out from the crowd. The natural choice. Everyone feels pleased with themselves and they throw a party.

Next, hardware. What machines does it run on? Well, only one actually. A large mainframe, usually available in a single colour, with an architecture that was designed in the early 1960s and remains largely unchanged to this day, and an operating system that is a proprietary hotchpotch of old systems tied together with bits of string.

The mainframe, bought in a hurry, is very expensive and no-one knows how it works but the package has to be loaded and a backlog of business is beginning to mount up on the old systems. The mainframe is delivered, installed and, in the well established tradition of have-a-go British middle management, the DP department has a go. The backlog continues. Things are not going so well.

In an important change of strategy, a professional manager from the package supplier is appointed to project manage the new system and run it on a day-to-day basis. That year, the DP budget soars from £1 million to £4 million. In the intervening period it also emerges that the package can't actually do the thing that it was supposed to do really well and, in trying to be all things to all men, it consistently delivers slightly the wrong thing. The tiny number of users who have been allowed to see it complain that the screens are cluttered with useless information and that it is impossible to see the wood for the trees.

Management are faced with some tough decisions, give up or keep going? They toss the same coin and it lands package side up *again*, so they decide to buy the source code. A few tweaks here and there and a place in the fast lane is assured.

To get the changes through in time, they employ contractors, loads of them, who are not exactly stupid and, knowing a good thing when they see it, make sure that they always come in on Saturday (double time, but you miss the football) and sit around reading *Viz.*

At this point it is instructive to look at the old systems, which are still running. They were written in undocumented spaghetti COBOL by a four man programming team two years ago, and were never designed to handle the work loads asked of them. They are beginning to fail on a regular basis but the business continues to rely on them and the users are beginning to treat the new system with something suspiciously like contempt.

The package project grinds on. At this point it gains a name, usually something from Greek mythology with letters forming a clever acronym. It's over-staffed, over budget and has overrun its schedules so often that everyone has difficulty keeping a straight face when another delay is announced. Slowly, the entire department becomes shackled to this monster. Work loads are reassigned, schedules reassessed, the specifications are changed - made larger and more woolly, to explain away the delays - and money is poured down its greedy throat in the hope that, well who knows? Anyone involved with the thing begins to experience a severe disorientation, similar to motion sickness, as the monster churns round and round.

No-one gets fired, just moved sideways. New management is brought in with inflated claims about potential cost benefits, upsizing, downsizing, anything. Careers are made and broken, babies born, ulcers grown, divorces granted, lives wrecked, all human life is here.

Simple, cost-saving projects are shelved permanently for lack of funds. Deadlines and milestones are missed, ass gets kicked (another US import), parts of the system stagger through quality control and are out-of-date on the day that they go live.

And the real joke amongst all this mess is that not a single penny has been spent on R&D into computing techniques. No-one knows a thing about new architectures and methodologies. Thirty years of academic research into computing and all that the DP department has managed to do is build a bonfire made out of 1960s architecture and COBOL and set light to it! They bloody well deserve everything they get.

Yo dude! Richard Samworth has attitude but works for a living at Software Components Limited as a software engineer. He can be contacted for an opinion any time on 0296 681 972. Yo! Get on down! (Wot 'Yo' mean? Ed.)

C LIBRARIES & TOOLS

COMMS LIBRARIES

C Asynch Manager V3s'ce	PC-DOS	£130
Essential Comms(L,MS,T)	PC-DOS	£170
Greenleaf Comms(source)	PC-DOS	£185
Multi-Comm (L,MS)	PC-DOS	£190
Silvercomm C Asynch	PC-DOS	£130

DATABASE/FILEHANDLING

Btrieve for OS/2	OS/2	£360
Btrieve for Windows	MS-DOS	£360
Btrieve v5.1	MS-DOS	£360
XQL	OS/2,PC-DOS	£440
Xtrieve Plus	OS/2,PC-DOS	£360
C/Database T'lchest sce	PC-DOS	£ 39
CBTREE	(source any C)	£135
C-Index/Plus	(source any C)	£175
C-ISAM (L,MS)	MS-DOS	£170
C-tree	(source any C)	£300
C-tree Plus	PC-DOS	£360
Essential Btree (s'ce)	PC-DOS	£105
CQL s'ce	PC-DOS	£270
Faircom Toolbox Prof v2	PC-DOS	£680
Pro-C & Workbench v2	PC-DOS	£585
R-tree	MS-DOS	£140
db-VISTA DBMS Object	PC-DOS	£450
Lattice dBCIII (L,MS)	MS-DOS	£185
Lattice dBCIII+	OS/2&MS-DOS	£365
SftFcus Btree&Isam	(sce any C)	£ 80

GRAPHICS

Essential Graphics v3	PC-DOS	£210
GFX Graphics	PC-DOS	£ 80
GraphiC v5 (CI,L,DS,MS)	PC-DOS	£245
graphics MENU/META v3	PC-DOS	£125
GSS Kernel system	OS2,PC-DOS	£425
GSS Graph Dev T'kit	OS2,PC-DOS	£425
GX Graphics	PC-DOS	£ 95
HALO v3 (MSC5,LAT,TC)	PC-DOS	£215
HALO Professional (C)	PC-DOS	£335
HGraph (MSC)	PC-DOS	£ 80
Ingraf (MSC)	PC-DOS	£185
MetaWINDOW v3.7b	PC-DOS	£145
MetaWINDOW/Plus	PC-DOS	£195
TurboWINDOW/C (Turbo)	PC-DOS	£ 85
QuickWINDOW/C (QC)	PC-DOS	£ 85

SCREEN & WINDOWS

C-Scape +Look&Feel	OS/2&PC-DOS	£390
Vitamin C for OS/2	OS/2&DOS	£240
C View Manager (s'ce)	PC-DOS	£245
Curses/PC w. s'ce	PC-DOS	£235
Entelekon Windows(s'ce)	PC-DOS	£105
Greenleaf Datawindow	PC-DOS	£205
Greenleaf Makeform	PC-DOS	£ 70
Lattice Curses V Lib	PC-DOS	£105
Multi-windows (MS,L)	PC-DOS	£190
Panel Plus II (source)	PC-DOS	£260
Power Screen (MS&TC)	PC-DOS	£ 99
Vermont Views (MSC,TC)	PC-DOS	£345
Vitamin C (source)	PC-DOS	£145

GENERAL FUNCTIONS

C/Utilities T'chest sce	PC-DOS	£ 39
C-Worthy w CV Architect	MS-DOS	£410
C Tools Plus/6.0 (MS)	PC-DOS	£ 99
CodeRunner	PC-DOS	£ 95
Entelekon Funct(source)	PC-DOS	£105
Essential C Utils v5	PC-DOS	£145
G'leaf Fin'cial Mathlib	PC-DOS	£205
G'leaf Super Functions	PC-DOS	£160
Greenleaf Functions s'ce	PC-DOS	£125
Lattice C Comp Kit+s'ce	PC-DOS	£160
PDK£1	PC-DOS	£ 70
Power Search	PC-DOS	£ 99
Resident-C (MS,L)(s'ce)	PC-DOS	£135
Silverware C EMM Lib	PC-DOS	£105
The Heap Expander(MS&TC)	PC-DOS	£ 65
Turbo C Tools v2	PC-DOS	£ 95
WKS Library	PC-DOS	£160

MATHEMATICAL LIBRARIES

C Lang Sci Lib v2	ANY C	£160
Mathpak 87 (L,MS)	MS-DOS	£ 80
Science & Eng Tools	MS-DOS	£ 59

TOOLS & SUNDRIES

PC-Lint	OS/2,MS-DOS	£ 80
C-Documentor	PC-DOS	£195
C-Scan	PC-DOS	£195
Clear+ for C	PC-DOS	£140
Databoss (C)	PC-DOS	£320
Lattice Comp Companion	MS-DOS	£ 70

REAL TIME & MULTI-TASKING TOOLS

Concurrent C (PC/MPX)sce	MS-DOS	£ 60
Multi-C (L,MS,CI,TC)	PC-DOS	£190
Op.Sys.T'box (MS) s'ce	PC-DOS	£ 69
Timeslicer v6 (MS6)	PC-DOS	£200
Over-C (L,MS)	PC-DOS	£225

ASSEMBLERS

At last, the new V6 of Microsoft's Macro Assembler is shipping. Call for details.

2500AD 80386 ASM v5	PC-DOS	£230
MS Macro-86 v5.1	OS/2&PC-DOS	£ 76
▶ **MS MASM PDS v6**	**OS/2&PC-DOS**	**£ 63**
PharLap 386 DOS Ext SDK	MS-DOS	£380
PharLap 386 ASM/Linkloc	MS-DOS	£835
PharLap 386/VMM	MS-DOS	£200
Optasm	MS-DOS	£110
▶ **Turbo Debugger v2**	**PC-DOS**	**£ 89**
2500AD Z80 ASM	CP/M-80	£ 70
SLR Z80ASM	CP/M-80	£ 40
SLR Z80ASM-PLUS	CP/M-80	£140
SLR MAC	CP/M-80	£ 40
SLR MAC-PLUS	CP/M-80	£140
SLR 180 (Hitachi)	CP/M-80	£ 40
SLR 180-PLUS (Hitachi)	CP/M-80	£140

C++ COMPILERS

Guidelines C++	PC-DOS	£370
▶ **Turbo C++**	**PC-DOS**	**£ 48**
Borland C++	PC-DOS	£206
Zortech C++	PC-DOS	£120
Zortech C++ DOS Dev	PC-DOS	£270
Zortech C++ OS/2 Dev Ed	OS/2	£360
Zortech C++ 386 Dev Ed	PC-DOS	£540

LIBRARIES & SUNDRIES

C++/Views (ZC++)	PC-DOS	£270
Greenleaf Comms (T&ZC++)	PC-DOS	£185
Zinc Interface Lib (C++)	PC-DOS	£150
Zortech C++ Database	PC-DOS	£180
Zortech C++ Video Course	VHS	£270

THE C LANGUAGE

New Topspeed C & C++ V3 now available.
New Hi-Tech C 386 PDS generates DOS and 386 code, includes own DOS Extender.

C COMPILERS

Zortech C++ OS/2 Dev	OS/2	£360
▶ **Microsoft C v6**	**OS/2&PC-DOS**	**£222**
Topspeed C for OS/2	OS/2	£335
Watcom C v8 Prof	OS/2&DOS	£310
Avocet AVCOM 86C	PC-DOS	£210
Aztec C86 Professional	MS-DOS	£120
Aztec C86 Developer	MS-DOS	£175
Aztec C86 Commercial	MS-DOS	£250
▶ **Borland C++**	**Windows/PC-DOS**	**£206**
HIGH C v1.61	MS-DOS	£390
HIGH C 386 Global	PL386&MS-DOS	£650
Hi-Tech C	MS-DOS	£160
Hi-Tech C 386 PDS	PC-DOS	£350
▶ **Microsoft QuickC v2.5**	**PC-DOS**	**£ 50**
▶ **M'soft QuickC/QuickAsm**	**PC-DOS**	**£ 83**
MIX C	MS-DOS	£ 20
MIX Power C & lib s'ce	PC-DOS	£ 30
▶ **Turbo C++**	**PC-DOS**	**£ 48**
Topspeed C Std	PC-DOS	£110
Topspeed C Prof	PC-DOS	£165
Topspeed C++ Std	PC-DOS	£110
Topspeed C++ Prof	PC-DOS	£165
Topspeed C++ Pro Dev Sys	PC-DOS	£220
Watcom C v8 Std	MS-DOS	£265
Watcom C/386 Std	PL386&MS-DOS	£585
Watcom C/386 Prof	PL386&MS-DOS	£780
Zortech C++ v2.1	PC-DOS	£120
Zortech C++ 386 Dev	PL386&DOS	£540
Avocet C Avcom Z80	CP/M-80	£210
Aztec C Personal 1.06D	CP/M-80	£120
Aztec Commercial 1.06D	CP/M-80	£200
Hi-Tech C	Z80 + CP/M-80	£105
Mix C	Z80 + CP/M-80	£ 35
Aztec C68K MPW C	MACINTOSH	£110
Aztec C68K C	MACINTOSH	£ 75
Aztec C68K/AM Developer	AMIGA	£175
Aztec C68K/AM Prof'snl	AMIGA	£120
Aztec C68K/ST Developer	ATARI	£175
Aztec C68K/ST Prof'snl	ATARI	£120
Hi-Tech C	ATARI	£105
Mark Williams C v2	ATARI	£110
Laser C	ATARI	£135
Prospero C	ATARI	£ 80

C CROSS COMPILERS

We supply 2500AD, Avocet, Aztec, Lattice, IAR and Hi-Tech Cross Compilers hosted on MS-DOS and targeted on Z80, 6502, 6801, 68HC11, 6301, 6809, 7811, 8051, 8096, 68000 & 68020. Please call for information or advice.

C INTERPRETERS

C-terp V3.0X	PC-DOS	£165
Instant C/16M	PC-DOS	£700
Interactive C	PC-DOS	£195
Introducing C	PC-DOS	£ 85
RUN/C Professional	MS-DOS	£110

PRICES & DELIVERY

Prices do not include VAT or other local taxes but do include delivery in UK and Europe. Please check prices at time of order, ads are prepared some weeks before publication. This page lists some of our products. Call us for a complete pricelist. Order by phone with your credit card.

LOW PRICES

FOR MICROSOFT & BORLAND PRODUCTS

WITH FULL TECHNICAL SUPPORT

Microsoft C PDS 6.0	222.00
Microsoft Basic PDS 7.1	222.00
Microsoft C/Windows SDK	364.00
Microsoft Cobol 4.0	365.00
Microsoft Fortran 5.0	205.00
Microsoft Macro Assembler 6.0	63.00
Microsoft Pascal 4.0	139.00
Microsoft QuickBASIC 4.5	50.00
Microsoft QuickC 2.51	50.00
Microsoft QuickC & Assembler	83.00
Microsoft QuickPASCAL 1.0	50.00
Microsoft Windows 3.0	70.00
Microsoft Windows SDK	265.00
Borland C++ 2.0	206.00
Turbo C++	48.00
Turbo Debugger & Tools 2.0	89.00
Turbo Pascal 6.0	69.00
Turbo Pascal Professional 6.0	137.00
Turbo Pascal for Windows	103.00
Object Vision 1.0	69.00
Paradox 3.5	409.00
Paradox Engine	223.00
Paradox MultiPack	615.00

CROSS ASSEMBLERS

We supply cross-assemblers by Avocet, 2500AD,Crash Barrier (METAI),IAR Systems and Pecan hosted on MS-DOS, CP/M-86 and CP/M-80 with over 30 target processors. In total over 200 products with no space to list them here. We hold some stock but you should allow 10-14 days for delivery.

PROGRAMMING TOOLS

Ada Compilers
Algol Compilers
Assemblers & Libs
AWK
Basic Compilers
Basic Interpreters
Basic Utilities
Basic Libraries
BCPL Compilers
C Compilers
C Interpreters
C Libraries
C Utilities
Cobol Compilers
Comms.Libraries
Cross Assemblers
Database Libs.
Debuggers
Dis-assemblers
Editors
Engineers Libs.
Expert Systems
Forth
Fortran Compilers
Fortran Libraries
Graphics Libraries
Icon
Linkers/Locaters
Lisp
Modula-2
Nial Interpreters
OPS 5
Pascal Compilers
Pascal Libraries
Prolog
Rexx
Risc
Screen Libraries
Smalltalk
Snobol

We stock many items for which there is no space in these advertisements.

DISK COPYING SERVICE

We can copy files to and from 600 disk formats including CP/M, CP/M-86, MS- DOS, PC-DOS, ISIS, APPLE, SIRIUS, BBC, TORCH, APRICOT, HP-150, TRSDOS, DEC RT- 11, IBM BEF, ATARI ST, AMSTRAD, MACINTOSH.

Our charge is £10.00 + disk + VAT with discounts on small quantities and disks are normally despatched within 24hrs of receipt. For more information call us.

GREY MATTER
2 Prigg Meadow, Ashburton, Devon TQ13 7DF
TEL. (0364) 53499

GREY MATTER
2 Prigg Meadow, Ashburton, Devon TQ13 7DF
TEL. (0364) 53499

GREY MATTER
2 Prigg Meadow, Ashburton, Devon TQ13 7DF
TEL. (0364) 53499

CIRCLE NO. 842

New C++ to C
Intek has released a C++ to C translator to compete with the Glockenspiel range. The DOS version runs in protected mode, and can support a number of MS-DOS C compilers, including Borland, Microsoft, Watcom and MetaWare. All memory models with their corresponding keywords (far, huge and so on) are supported. Intek costs $495, and is available from Intek Integration Technologies (0101 206 455 9935).

Codebase++ for Windows
Sequiter has released a DLL version of Codebase++, the C++ dBASE-compatible database manager package. The DLL can be shipped free with developed applications. Note that, unlike other DLLs, which you can generally use with any old language, name manglement means you won't be able to use this DLL with anything but Borland C++. Codebase++ V1.01 costs £190 with source, from The Software Construction Company (0763 244114).

Paradox for Pascal, Windows
The long awaited new version of the Paradox Engine is now available, with a DLL for Windows language and TPU for Turbo Pascal fiends. The price is £299.95 - V1.0 owners can upgrade for £69.95. Borland is on 0734 320022.

Sales channel
Channel Business Systems is compiling two disk-based catalogues of British PC software. One will list available commercial software, the other shareware. The disks will be sold for around £6. The company is keen to contact authors who wish their products to be included - call CBS on 0843 299597 with your details. The company will also manage registration and first line support for shareware authors.

UNIX for jeans
American company SSC is the first on the block to release a pocket reference to UNIX System V.4. Its C Library Reference is 72 pages long, and includes all the new system calls (virtual file systems, multiple processors, the programmable scheduler). ANSI prototypes are provided, and ANSI standard library calls are labelled as such. Telephone 0101 206 527 3385 for details.

Cocklecarrot presiding
Judge Hatter has changed his mind on the apparently-resolved Ashton-Tate vs Fox; Ashton-Tate **does** *own the copyright to dBASE, after all. Fox is back in the ring - the case continues...*

Grandmother sucks eggs

Micro Focus and Microsoft have launched a posse of new products to bring COBOL kicking and screaming into the present. Their announcements cover the juicy new worlds of SQL, GUIs, Windows and OOP.

The first is a GUI painter/application generator for COBOL, which was launched at last month's Which Computer Show. Dialog System V2.0 produces code which can run windowed applications under DOS, OS/2 character mode and Presentation Manager. It supports CUA/SAA, and produces code compatible with Micro Focus's COBOL V2.4 and its close cousin Microsoft COBOL V4.0.

Dialog System isn't available for Windows yet, so in the meantime COBOL users keen to knock up a Windows application should obtain the free 'Windows Enabling Kit' from Micro Focus. This allows the compilers to generate Windows prologue and epilogue code, with support for Windows API calls via the Windows SDK. Micro Focus is on 0635 32646.

Microsoft and Micro Focus users can now also embed SQL in COBOL source, with Microsoft's Embedded SQL for COBOL. This is a preprocessor which converts embedded code into the appropriate database calls to Microsoft's SQL Server. It includes cursor support, and ANSI and SAA standards for embedded SQL.

As for OOP - well, it's not quite COBOL++, but Micro Focus has recently signed an agreement with Digitalk that will allow Smalltalk/V PM to be integrated into the COBOL/2 Workbench. The Workbench will provided a messaging interface between Smalltalk objects and COBOL applications. The product is still in the pipeline, so it's too early to tell whether this is a cunning hybrid or a bit of bodge. We'll keep you posted.

Paradox code generation

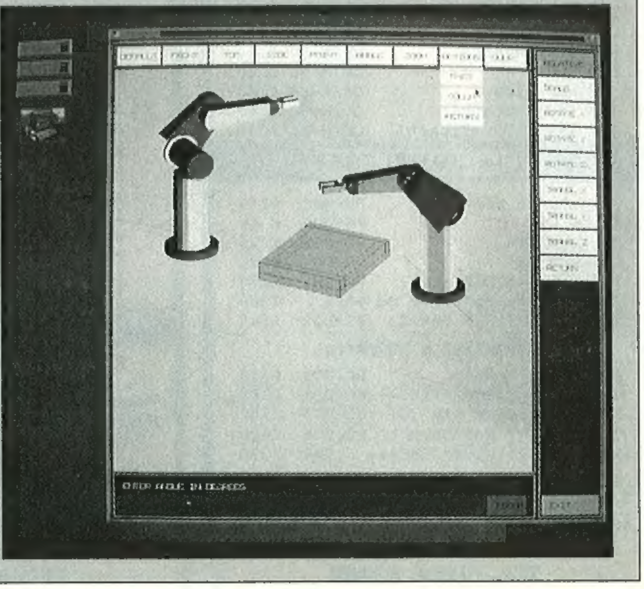
PARAGen is a code generator that will produce Pascal, C, and C++ (no OOP) source code compatible with the Paradox Engine, based on your existing Paradox tables. You just select the Paradox file, and pick out the combinations of fields and Engine functions you want. The code is well-commented, and you can generate code optimised for size or speed, manipulate bracing style, include error checking and so on. PARAGen costs $129 and is available for DOS and Windows on 0101 219 397 8952.

Workstation standards on PC

The oozing together of the workstation and 386 PC market continues. This month two processor-intensive graphics standards normally associated with workstation applications made the crossing to the PC.

Template Graphics Software (071 799 2434) has announced a version of PHIGS for 386 UNIX systems with X Windows. PHIGS is the Programmers Hierarchical Interactive Graphics Systems, an ANSI/ISO-proposed API for dynamic modelling, viewing, editing and storage of 2D and 3D objects - traditional workstation territory, in other words. The pack, called FIGARO+, costs £1100; as well as implementing the standard PHIGS bindings (object construction with primitives, database maintenance), it includes some of the PHIGS+ extensions for lighting and shading realistic models. Eight megabytes and a coproc are recommended, say TGS.

In the same vein, Scientific Software Ltd has announced a protected mode version GKS (the other ISO graphics standard for 2D designs). The package runs under the NDP compiler, and costs £595. A real mode GKS for Microsoft C is also available for £395. For those who prefer FORTRAN, Salford has released a similar bundle for its 32-bit FORTRAN compilers, called GKS/386, and priced at £425. Salford is on 061 745 5678; Scientific is on 0628 890011.



Rainbow Technologies -
Leading the world in software protection

EYE™
RAINBOW TECH, INC.
Sentinel Pro-8b
RAINBOW TECHNOLOGIES, INC.
NetSentinel
RAINBOW TECHNOLOGIES, INC.
SentinelPro™
RAINBOW TECHNOLOGIES, INC.
SentinelPro
RAINBOW TECHNOLOGIES, INC.

With more than 1.5 million Rainbow keys protecting over $2 billion worth of software and a customer base of over 5,000 developers. Rainbow has gained more experience in Execution control, Revenue protection and Information security than any other company.

Rainbow keys offer virtually unbreakable security against unauthorised software use. Custom designed using VLSI and ASIC technology, we have a whole family of Hardware keys, with over 100 high-level language interfaces available for most operating environments, including DOS, OS/2 Windows, Unix, Xenix, Novell and Apple Macintosh.

Worried about customer acceptance?

Rainbow keys are easy to install and totally transparent in operation. Unlimited backup copies can be made (the application will only run when the key is present in the parallel port). A recent survey by Chris Naylor in *PC Week* said *"Rainbow's Sentinel was so easy to set up and to use that compared to some models, it almost felt like cheating"*

We are not the cheapest...

You get what you pay for. We supply a very high quality product, engineered to exacting standards of quality and reliability and backed by the very best in World-class customer support. So when you buy Rainbow, you are not simply buying the latest technology but the support of a team of experts unrivaled anywhere in the World.

That is why many of the best known names in Software Development use Rainbow Keys...

Rainbow Technologies spends far more on Research and Development than any of our competitors. We believe in offering a tried and tested, proven product, which does what it was designed to do, reliably and at a fair price.

Software Developers

Protect your sales and profits with Rainbow's Software Sentinel keys...

- **Sentinel Pro™**
  Rainbow's flagship key uses a variable algorithm technique for maximum security.
- **Sentinel C™**
  Protects multiple packages using developer programmable memory.
- **Sentinel Scribe™**
  Version of Sentinel C which can be dynamically alterable in the field.
- **Sentinel Shell™**
  Protects without requiring access to the source code.
- **Net Sentinel™**
  Site licence revenue protection of LAN applications (NetPro & Net-C).
- **EvE™**
  For Apple Macintosh SE and II, protects Multiple packages, uses ADB port.

Trade marks where used are acknowledged.

RAINBOW TECHNOLOGIES

Call 0753 41512 today for your Free 28 day trial

or fill in the coupon and post it to Rainbow Technologies Ltd., Shirley Lodge , 470 London Road, Slough SL3 8QY

Name ....

Title ....

Company..

Address ....

....

Post Code ....

Tel: ....

Yes, I would like the opportunity to evaluate a Rainbow Security Key. I am interested in:
(tick box)

❑ Sentinel Pro™ ❑ Sentinel Shell™
❑ Sentinel C™ ❑ Net Sentinel™
❑ Sentinel Scribe™ ❑ EvE™

Rainbow Technologies reserves the right to refuse participation in this promotion

.EXE 6/91

CIRCLE NO. 843

Cheap 8514/A

The WIZZ graphics card is a fast, low-cost version of the IBM 8514/A, suitable for PS/2 users needing a second monitor with Windows. The maker, Tektite, say that WIZZ is fully compatible with both the 8514 and VGA standards while costing only £195 - around a fifth of IBM's price. Tektite is on 0394 672117.

New Asymetrix ToolBook

V1.5 of ToolBook, the Windows development package, is now available from NEOW. Improvements include a noticeable notch-up in speed, larger script and bitmap handling, importable graphics, and DLL/DDE support. Extra default objects and messages are also available. Toolbook V1.5 costs £310, with upgrades for £75. NEOW is on 0628 668334.

Going down

Borland, following its announcement that users of its C++ products now number over 350,000, has dropped Turbo C, and lowered the price of Turbo C++ to £69.95 to take its place as the budget package. Borland is on, as ever, 0734 320022.

C-Worthy upgrade

Fans of C-Worthy, Solution Systems' text screen library for Microsoft and Turbo C, should note that there is a major upgrade in the offing: new features include an improved cwArchitect screen designer which allows dialog changes to be made on the fly, redefinable colour palettes, CUA compliance and .PCX file support. C-Worthy V2.0 costs £559 with source, £389 without. Upgrades are £175 (with), £125 (without).

UPS size down

The Avel MP8 is a new uninterruptable power supply that costs less than a grand. It's also rather titchy for an UPS, only 20x45x26cm. Avel claims the system can isolate systems from all mains disturbances as well as providing at least eight minutes of 240V, 50Hz power in the event of a power failure. Audible and visual alarms show the how much power is being used, and how much is still available. Avel is on 0708 853444.

Easy virtual memory

George Consultant's STEM is a new virtual memory handler that provides a simple C interface for all the storage space on a PC : real, expanded, extended and disk. It costs £195, is available for Zortech, Microsoft and Borland Turbo C, will run on any PC, and has been, says George, extensively tested. Telephone 071 376 8767 for details.

Tap into America

Internet is the high-speed TCP/IP network that carries mail, data (and the occasional worm) between academic, research and commercial sites in the US. Many US companies use it to distribute upgrades to their customers - Sun has just placed the entire source to its transport independent remote procedure call interface onto it, and the throughput of free high-grade information and development software is considerable. The C++ and MS-DOS newsfeeds easily top 3 or 4 MB of information and code a week.

Sadly, UK systems have only been able to connect with the Internet via the slow and fragile UUCP store-and-forward mail system. Messages can get lost and interactive computing is impossible. A pukka high-speed link-up between UK sites and the States requires a great deal of money and organisation. The recently formed United Kingdom Internet Consortium intends to be that organisation ; it's a non-profit company composed of individuals, companies and user groups who plan to have a UK Internet up by the end of 1992. The UKIC is currently running trials; companies interested in joining the 'net as a founder subscriber should contact them on 071 269 4159 or write to the UKIC, PO Box 360, HARROW, HA1 4LQ.

Nice book...

'Oh My! Modula-2' is the teeth-gritting title of an otherwise excellent package - a 750 page introduction to programming that includes a full Stony Brook QuickMod DOS compiler on disk. While it's aimed at the academic market, £25 is not a price to be sniffed at compiler-wise, and the importers, Real Time Associates, say it should be available in major bookshops soon. Support, upgrades and registration are not available for compilers purchased via 'Oh My!'. RTA is on 081 656 7333.

Binding agreements

Phar Lap has answered criticism over its strict run-time licensing policy with a new deal for 286 | DOS-Extender users.

Phar Lap is now offering a 'one-time' license service, costing £615, which allows unlimited distribution of a single bound product. The Phar Lap banner has to be displayed within the program. A more extensive licensing agreement, with various warranties and the option not to have Phar Lap plastered everywhere, begins at £3100 for the first 2000 copies. Phar Lap's distributor in the UK is System Star: 0992 500919.

New Clipper

Nantucket has released an update to Clipper V5.0, to be mailed automatically to registered UK Clipper users. The most important change is to the V5.0 virtual memory manager, which has been refined (Nantucket's term) to stop it dying so painfully in memory shortages. Summer '87 compatibility has been tweaked up too. The Clipper debugger has been entirely done over, with improvements to script support and a nifty new object browser. Elsewhere, runtime messages are a bit less arcane, more `get` class instance variables are made public, and colour support has been improved.

Most interesting (but not overly crowed about) of the V5.01 features is the new `USE VIA <F>(cDriver)` command. This will allow Clipper programmers to use standard syntax to access databases other than the xBASE .DBF format, using specially written device drivers. Don't hold your breath for the drivers, however ; like V5.01's halfway objects (see 'Objects for Clipper'), Nantucket has only ensured that Clipper can support `USE VIA` in the future. It has yet to release any drivers itself or an SDK for third-party developers. Something to watch for in the next few months, perhaps. Nantucket is on 0707 373600.

When Clipper upgrades...

Blinker users will need to obtain a new copy to cope with the new V5.01 Clipper. Unfortunately, many users have forgotten to post their registration cards, and may miss out. If you aren't sure whether you are registered or not, phone the distributors QBS on 081 994 3441. A good test of registree status, says QBS, is if you've ever received a copy of QBS Software News. If you haven't, please get in touch.

Pocket-priced UNIX

Coherent and MINIX, the two low cost UNIXs we occasionally harp upon, both now have UK dealers. Coherent, reviewed in November 1990, is to be sold by AET of Lichfield, Staffordshire (0543 416550), priced £99.95. The MINIX Centre in Norfolk caters for Brit MINIXers. On top of standard Prentice Hall MINIX V1.5 for the PC, ST and Amiga, the Centre also sells its own adapted PC version which operates in 386 flat-memory. It costs £106, together with a port of the protected mode GNU C compiler. The MINIX Centre is on 0953 789345.

Fit Borland into your programme...

on Stand No. 59 at the Software Tools Show

11–13th June 1991 at Wembley Exhibition Centre

The makers of Paradox,® Quattro® Pro, Sidekick,® ObjectVision,® Turbo Pascal,® Turbo Pascal for Windows,® and Borland C++®

BORLAND
SOFTWARE
KNOW HOW

Please rush me a FREE information pack on:

Turbo Pascal □ Turbo Pascal for Windows □ Borland C++ □

Another Borland product ______________________ (Please state)

NAME ______________________ JOB TITLE ______________________

COMPANY ______________________

ADDRESS ______________________

POSTCODE ______________________ TELEPHONE ______________________

EXE 6/91

I currently use Borland's ______________________ (Please state)

Another language ______________________ (Please state) I currently use DOS □ Windows □

Please return this coupon to: Borland International (UK) Ltd, FREEPOST, PO Box 21, Aylesbury Road, Thame, Oxon OX9 3BR.

CIRCLE NO. 844

OOP tutes
Transcripts of the proceedings and tutorials of TOOLS '91, the OOP conference held in Paris this March, are now available. Topics covered include Adele Goldberg on OO Project Management, Steve Cook's comparison of OOP languages, OO Structured design, and introductions to OOP databases, Smalltalk, Beta and Eiffel. More information and order forms can be obtained from Applied Logic on 081 780 2324.

Unfair exchange?
In the front of the telephone book is a list of STD codes, sorted by town. This is free to all telephone owners. Some people would find the same list sorted by number more useful; a book with this in costs £4.50. Some developers would like to include the whole database in electronic form. This costs £999. The STD Code Decoder Developers' kit is available from the WSA Consultancy on 0992 553341.

Cameo appearance
Noble Campion holds the brains behind the Pluto graphics board of a few years back. These days, it generally does bespokes graphics systems for big clients. Now it's put together some of its processing, format conversion and output expertise into a PC command line program called Cameo Image. Chief features are its extensive plotter and display support and extensive palette manipulation, including 24-bit colour pictures and transparency simulation. Noble Campion is on 081 951 5656.

Confounding CONFIG
Opensoft has announced Memory Commander, a TSR and device driver highloader that does not require CONFIG.SYS or AUTOEXEC.BATs to be changed. It can also speed up BIOS operations by transferring the ROM BIOS into RAM, if your machine doesn't do that already, and emulates XMS and EMS. Memory Commander is available from OpenSoft for £49.

M/4 for Windows
Micro Data Base Systems Inc has, via its UK distributors Panelhigh, announced the release of M/4 for Windows. The software includes a data dictionary, API, developer tools and 90 days of technical support. Add-ons include a transaction processing API offering transaction logging, roll-back and -forward, abortable transactions and fault tolerance. M/4 also works with Object/1, mdbms' OOP development environment for Windows. M/4 lists at £750 - telephone Panelhigh on 081 903 0211.

Extra Windows info

Microsoft US, clearly aware of the current paucity of Windows information, has published a 200 page Windows Resource Kit. The kit includes technical material that has been produced since the last release of Windows. Topics covered are more detailed descriptions of configuration, DOS boxes, PIF files and networking, as well as an index of sources of further information, such as user groups, books, journals and on-line listings. The kit also includes several utilities, including the hDC Memory Viewer and a screen saver.

Dapper KAPPA

Demonstration disks are generally a waste of time - sales puff and screenshots stapled together with 'Press Spacebar for More' messages. Intellicorp's demo for KAPPA-PC is rather different, and has had me fascinated all month.

It's a cut down version of Intellicorp's Windows development environment, a graphic (but C like) object oriented interpreter with a Prolog-like inference engine. Sounds terrifying on paper, and the sales literature screenshots jam-packed with icons and trees don't help. All, however, becomes clear when you run the demo disk.

The demo is a full implementation of KAPPA-PC, together with a written tutorial. This takes you through designing a class and instance hierarchy, building up methods, adding goal-based reasoning and constructing the user interface until you have a full application. The demo is crippled only in that you have limited resources - 10 classes, 25 instances and so on. It also omits the final versions ability to integrate C code into applications (presumably through DLLs). Other than that, you can play to your heart's content. The ease of use of the interface sells itself; I've read countless articles on Prolog, but this was the program that gave me a real feel for backchaining. Environments as easy to use as this are currently rare and expensive - KAPPA-PC itself costs £2,500 - but it's the stuff of the future. Which makes this sneaky cheap preview all the more interesting. The free Ten Step Demo can be obtained from IntelliCorp on 0962 73548.

$25 Network

We've always resisted the temptation to install a LAN at the .EXE offices because of all the horror stories that we hear. The editorial department has put up with SneakerNet (a snazzy American way of saying that we swap floppies by hand) for so long, though, that some automation is now definitely called for.

The $25 Network could be the solution. This ingenious package links two PCs via their RS-232 ports. The only hardware involved is the cable. The software consists of a device driver which routes calls to external drives across the cable to the other machine, and all this can take place while the other machine is in use. We've seen the system in action and, while it's nowhere near as fast as a 'real' network, it does allow easy access to one machine from another.

UK price is £25, details from EQ Consultants.

Stop Press

Late news, just in. Our Xbase correspondent has rushed in from the FoxPro 2.0 launch, dumped a press pack on the desk and disappeared again. Let's see what it says: Rushmore Query Optimisation... Relational Query By Example... .EXE compiler and an API... special 'extended-mode' 386 version... Fox claims it's very fast... ships in June, price £149. Hmm. Pity Ian didn't hang around to give us his own impressions. Hold up, he's written something on the envelope: *'Fast - or What?'* Fox Software is on 0462 421999.

User objects for Clipper

Nantucket has a softly, softly approach to OOPs: Clipper V5.0 had a few fixed objects (notably the TBrowse database browser) and the syntax to support them but did not allow the user to create his own. Nantucket's plan, one supposes, was to get XBasers a sniff of the brave new world, then sweep in with an irresistibly tasty all-OOP-action upgrade.

Internal groundings for objects were finalised in the new Clipper V5.01 (see 'New Clipper'), but Nantucket has been beaten to User Defined Objects by ChyDale, Clipper utility merchants from Bishops Stortford. ChyDale's UDOs use the Nantucket syntax and compiler hooks to allow users to write their own objects with instance variables (public and private), methods and constructors. The package includes samples of menu handler, stack, linked list, dictionary, image and time objects. Chris Sennitt, founder of ChyDale and previously Nantucket Europe's Technical Director, also claims to have implemented single and multiple inheritance. ChyDale's User Defined Objects sell for £159; telephone 0279 758022 for details.

Software, Beer and San Francisco

Just back from a week-long Borland languages conference in San Francisco (pause for cries of 'Spawny Gett!' from jealous readership). Surprisingly, Borland itself had comparatively little to announce. Its Turbo Pascal class libraries Turbo Vision (MS-DOS based) and Object Windows (Windows based - not to be confused with Object Vision, which is Borland's Windows 4GL product) are to be ported to C++ 'in the not too distant future'. Less predictably, Borland announced that it was working on an XBase development environment ('XBase' is the new jargon for what used to be called 'dBASE-compatible'). The XBase product will support Paradox and XBase data files, support the XBase language and run under Windows. Borland says this will appear within the next 12 months.

Rogue Wave

At a third party vendors' exhibition at the conference, the ground was so thickly covered with class libraries for C++ and Turbo Pascal that one could hardly turn around without treading on one. Rogue Wave, whose C++ Foundation Class Library *Tools.h++* is to be bundled with the elusive TopSpeed C++, was showing V4.0 of that product. Tools.h++ is (uniquely, according to RW) compatible with Windows 3.0, and includes classes for DDE, Clipboard stream buffers, regular expressions and tokenisers as well as the familiar strings, dates, linked lists etc. The library supports all known MS-DOS compilers plus some UNIX ones, and costs US $199 for MS-DOS with source, or $99 for object code only. The company also offers a maths class library *Math.h++*, and can be contacted on 0101 503 745 5908.

Zinc

Last month, we wrote (in a review of Turbo Pascal for Windows) that it was impractical to combine a DOS and Windows GUI class library so that source could be ported from one to the other. Zinc Software has gone out of its way to make us look foolish in double-quick time. V1.0 of the Zinc Interface Library supports a text-mode and graphics-mode GUI from a single MS-DOS executable. V2.0, which should come out by the beginning of June, lets you produce dual-mode DOS/Windows applications at the cost of a single `#IFDEF` directive at the top of your code. A system of persistent objects allows screens to be designed in DOS graphics mode, then imported into Windows applications (or, in a simplified form, into DOS text-mode applications). Zinc supports Zortech and Borland C++ (separate versions) and is priced at $199.95 (DOS only) and $299.95 (DOS and Windows). Zinc is on 0101 801 785 8900; or, if you prefer a UK contact, try Zortech (081 316 7777).

TEGL

TEGL Systems' Richard Tom got into the business of devising go-faster graphics in order to support the games that he was writing. He claims that the BGI-compatible TGI graphics library in the TEGL Windows Toolkit runs twice as fast as Borland's offering (TEGL's library is written in assembler and incorporates dirty tricks such as accessing EGA hardware direct). My impression, from watching a demo of his graphics-mode windowing system - including a silk-smooth sideways scroll - is that this is correct. As well as the graphics library, TEGL offers a very slick Windows-like event-driven GUI system (complete with editable icons, timer events, menus etc) and a virtual memory manager. Surprisingly, the API is not yet object-oriented, but Mr Tom is working on it and will deliver a free upgrade. The system supports Turbo Pascal, QuickPascal, Turbo/Borland C++, Microsoft C and Watcom C. Price is $99 + $15 shipping including source, and for an extra $139 you can also have the source for all the games that Mr Tom wrote. TEGL is a Canadian company, phone no 0101 604 669 2577.

TesSeRact offer

Owners of Borland C++ can get a free copy of TesSeRact, the famous TSR development kit, straight from its US developers IDC simply by phoning the company up (0101 215 443 9705) and quoting their BC++ serial number. Offer ends July 1st. If you don't own Borland C++, you will find shareware versions of the library on various bulletin boards, and an early .EXE disk. It supports most C compilers. Incidentally, IDC produces the CXL user interface library that people ring us up about every 25 minutes (which suggests it must be quite good). Don't ring us, ring them!

TP6 Development Kit

Blaise Computing, like many others, has a Windows C++ class library (Win++, costs $249, supports Borland C++ only). However, this item celebrates the company's Turbo Vision Development Toolkit. You get a Resource Editor for creating dialog boxes, a utility to convert Turbo Vision resources into Windows resource script files and an object library which extends Turbo Vision's capabilities. The package costs $149, and is available from Blaise on 0101 415 540 5441.

Non-Cosmic Glock

Blaise's C++ library is what is known as 'cosmic', ie all classes are derived from a single 'cosmic' class in Smalltalk fashion. The boys at Glockenspiel, whose Commonview multi-GUI C++ class library is the grand-daddy of them all, scorn this approach, saying that it suggests 'intellectual poverty' in the design. Glock points out that it is very hard work to use more than one cosmic library in a single application. Non-cosmic class libraries such as, er, Commonview, are much more *interoperable* ie you can use them with anything you like. Glock's silver-tongued techie Fergal Dearle got me completely convinced of this over a sequence of glasses of San Franciscan Anchor Steam Beer, but when the morning came, all the clever details had eluded me... Glockenspiel had Windows and OS/2 PM versions of Commonview ready many moons ago, and is now starting to release OSF Motif variations for various UNIX platforms. To find out if it has reached your platform, call 010 3531 733166, or UK distributor QA Training on 0285 655888.

Draft Dog

Total self-indulgence: fellow drinkers of Newcastle Brown Ale occasionally fall into huge mega-arguments about whether the Northern nectar is available on draft. I can report that a bar in San Francisco has a draft tap bearing the famous figure-of-eight badge, but when I dragged the greater part of the UK press corps along to sample it, it turned out to be cola-coloured fizzy water. Note to S&N Breweries: in far-off lands your name is being taken in vain.

Workstations from Digital.

To get the most out of your applications ...

Letters

We welcome short letters on any subject that is relevant to software development. Please write to The Editor, .EXE Magazine, 10 Barley Mow Passage, Chiswick, London W4 4PH. Unless your letter is marked 'Not for Publication', it will be considered for inclusion on this page.

A letter to Verity

Dear Verity,

Re: Project 3 (STOB, .EXE Feb '91). I once had a portable Osborne micro returned to me with the remains of a cup of sticky coffee gumming up the keyboard. It was urgently required for a class that afternoon. All attempts to revive it failed. It eventually went under the tap and was thoroughly swilled and drained. A couple of hours behind my office radiator completed the treatment. It was restored to its former capability.

As you were going to order a new keyboard anyway - it's worth a try.

M Finean
Aston University
Birmingham

Harry trouble

Sir,

I have just discovered a bug in the Microsoft V6.0 Harry (Harry Secombe-Piler, geddit?) which you may be interested in. It can result in very obscure and potentially dangerous bugs.

```
    int fred(int x)
  {
   switch(x)
   {
     case 300:
       return 1;
     case 20:
       return 99;
     case 21:
       return 99;
     case 22:
       return 99;
     case 23:
       return 99;
     case 24:
       return 99;
     case 25:
       return 99;
     default:
       return 600;
   }
  }
```

Figure 1 - MS C bug

It disappears with optimisation disabled but is present with the default level of optimisation. I have only tried it in large and small models. Please see Figure 1.

What happens is: the `default` statement is not executed if the switch variable `x` is greater than 300, it is executed if the value is less than 20 or between 26 and 299. Moving the `case` statements around makes no difference. You can see what's going wrong by compiling with the /Fa switch and looking at the assembler output.

I haven't informed Microsoft about this one, I can't be bothered spending a day waiting to get through to them on the phone and I don't know their fax number. I expect someone has already discovered it and told them, but I don't remember seeing this bug in .EXE - hence this letter.

Dave Riley
Tele Control Communication PLC
Basingstoke

From the other side

Sir,

Your report of my death (cover illustration, April '91) is greatly exaggerated.

P Collinson
pp Michael (Messy) S. Dos
Doschester

Make my Day

Sir,

With reference to John Barber's Soapbox spiel, I would just like to point him, and any other frustrated DOS Make users, at Sun Make. Ok, yes it is a UNIX tool, no it does not run under DOS, but who in their right mind would attempt serious programming under DOS?

(Software engineer, over 10 years experience, who remembers real computers, before DOS came along and set back software engineering several years).

PS. If you do look at SunOS Make, then you really MUST look at N.S.E!

PPS. Yes, I would like to join the 'Ban the PC' lobby.

AJ Rideout
Rosemount Ltd
Bognor Regis

A real character

Sir,

I read with interest the article by Ebbe Sønderby regarding the difficulty with PCs when using international character sets.

The THEOS multiuser operating system was designed from the outset to make it portable across different international languages, including those which require eight-bit characters sets, such as Spanish and Danish. Most of the international eight-bit character set is catered for by THEOS's printer drivers. In addition, THEOS is available in Japanese form, both Kanji and Katakana character sets.

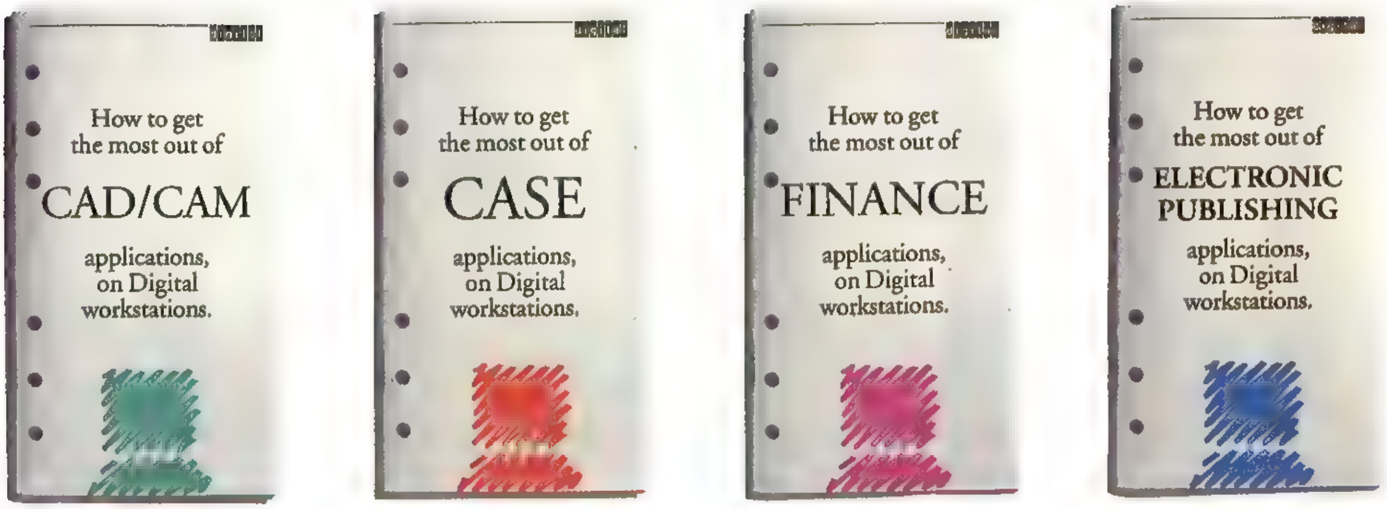
As all system messages are stored in text files, and each user may have their own command synonym file, it is possible for many different users on the same system to each use THEOS in a different language. A parameter in the main system configuration file, and a further parameter in each users LOGON parameter file, governs respectively the default system language code, and each individual user's language code.

Whilst it is true that several operating systems are available in foreign versions, it is most unusual to find an American system which has obviously considered the European market-place to this extent.

Mike Edwards
THEOS Product Manager
Midshires Computer Services
Cheshire

Letters submitted to this page may be edited. The writer of the best letter of the month, as judged by the Editor, will be rewarded by a T-shirt or similar-valued .EXE trinket. The best letter is the one printed first.

digital


How to get the most out of CAD/CAM applications, on Digital workstations.
How to get the most out of CASE applications, on Digital workstations.
How to get the most out of FINANCE applications, on Digital workstations.
How to get the most out of ELECTRONIC PUBLISHING applications, on Digital workstations.

Take guidance!

You've found your application ... you've seen it running ... and you know what it can do.

Just two more decisions to make ... what platform to use, and where to get support for the smooth running of the whole solution.

Digital makes these decisions easy. We've taken four of the most popular application areas – CAD/CAM, CASE, finance, and electronic publishing – and produced a free guide for each that covers the commonly used applications packages, and the machine and network configurations that will help you get the most out of them.

Specific topics include ...

- how to get your products to market *faster* with CAD/CAM, by integrating design engineers with the whole design team ...
- Digital's COHESION environment for CASE – for software that's successful, on time, serviceable to spec., and at the right price ...
- how finance systems can integrate applications to deliver intelligence right to the desk-top ...
- how Digital provides electronic publishing solutions that go beyond the desktop, covering the complete document management and distribution process.

Each guide covers the comprehensive support Digital offers, and is accompanied by information supplied direct by third parties.

See how they run on workstations from Digital

To match your choice of applications, Digital offers a choice of workstations – with a wide range of price/performance options. Whatever your application there's a Digital workstation to run it, from entry level VMS or UNIX® through to high-performance UNIX workstations, VMS and UNIX servers – all with unmatched capability for system expansion and integration into your enterprise.

And support? From Digital, it's comprehensive – through service, training and consultancy. In fact, support from Digital starts *before* you've even chosen a Digital workstation, with these four new Workstation Application Guides.

Send for your *free* copies – today!

Free, pocket-sized, packed with information – these are guides you should have *now*. Choose one, or ask for the whole set. For your copies post the coupon today.

To: Application Guides, Digital Equipment Co. Ltd,
PO Box 525, RHA-PR, Maidenhead,
Berks., SL6 1YU.

Please send me the application guide for:

☑ *(Tick)* □ CASE
□ CAD/CAM □ Electronic Publishing
□ Finance

□ Details of Digital's workstations range

I intend buying workstations in the next 6 months? □ Yes □ No

Name *(Mr, Mrs, Ms)* ______

Position ______

Organisation ______

Address ______

______

Postcode ______ Tel ______

Industry ______ .EXE 6/91

The Digital Logo, COHESION and VMS are trademarks of Digital Equipment Corporation. UNIX is a registered tradmark of UNIX Systems Laboratories Inc.

CIRCLE NO. 847

Playing the Generation Game

It's the future of 4GLs, not COBOL, which is threatened by the brave new world of 'application generation', believes Patrick McParland.

The arguments for application generation go something like this. Software engineering is taking software development from pencil-and-paper-based development and 3GL coding to automated software development. Application Generators are part of this. They are software tools which transform a specification into a ready-to-execute application. Developers who use an application generator, in conjunction with structured techniques, can gain significant improvements in their productivity, the quality of the completed application and in easier maintenance. As application generators become more powerful, and so bring further increases in productivity and quality, developers will have to abandon the 'old ways' of detailed coding and embrace the benefits of automated software development.

Until recently, the term *Application Generator* effectively meant little more than Fourth Generation Environments (4GEs) and code generators. Today that is very different. Many Computer Aided Software Engineering (CASE) tools now support the automatic generation of an application from a developer's analysis and design. And such CASE tools are now within the scope of application generation technology. At a stroke, the inclusion of CASE expands the scope of application generation technology: from a purely design and construction aid to the complete software development process. Reviewing application generators within this new context is the purpose of this article.

Classification

The most common classification of application generators divides them into *End-user* and *Development* generators. End-user application generators help novice developers or educated end-users to produce simple applications quickly or to customise existing applications. End-user application generators provide easy to use screen painters, report writers, database query languages and menu designers. Development application generators, on the other hand, are for the more expert developer who wishes to produce complex applications. A powerful programming language, or 4GL, is the important part of these generators.

Classifying application generators is made difficult by the fact that most application generators try to satisfy the demands of both novice and expert users, and fall between the two camps. They provide easy to use design tools and a 4GL. The design tools produce prototypes which developers can customise using the 4GL. Also, recent developments, such as the wider use of GUIs, have made it possible for application generators to provide more comprehensive support for all categories of user.

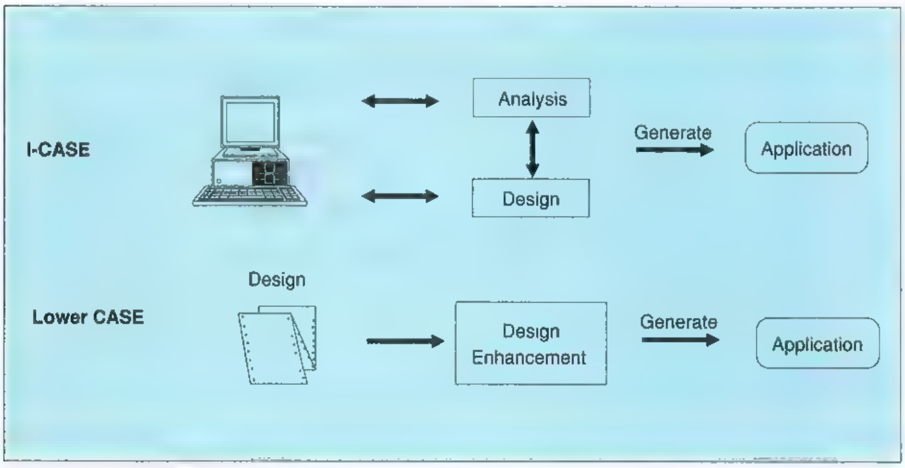


Figure 1 - Application Generator Classifications

A better way of classifying generators is to use the form of input given to them as the main distinguishing feature. There are two main kinds of input to a generator. One is the keystrokes of a developer which defines an application. The second is the output from another, high-level tool, such as an analysis and design package. The two types of classification for these are termed 'Lower CASE', where the developer drives the package direct, and 'I-CASE' where another package provides the input.

These classifications reflect the changes in technology which allow the generation of applications from analysis and design information. There is also a clear division between the classifications. I-CASE tools concentrate on using the analysis and design of an application to produce an application. Lower CASE tools allow a developer to input the design of an application, and to refine it into a completed application (Figure 1).

I-CASE

I-CASE generators consist of an integrated set of tools built around a data dictionary or repository. They work like this. Usually supporting one or more development methodologies, they provide diagramming tools to allow a developer to specify an application using analysis techniques such as Entity Relationship modelling, Data Flow Diagrams and Structure Charts. These diagramming tools populate the tool's data dictionary with a model of the application. Then design tools are used to produce a more implementation-specific model of the application by incorporating more detail into the Entity Relationship models, the Data Flow Diagrams and others. For detailed design, an I-CASE tool may also provide a pseudo-code editor and tools to design screens, reports, menus and database queries. The I-CASE products use their generation tools to generate a completed application (Figure 2).

I-CASE generators advocate a top-down approach to software development. In this approach, the analysis and design of the application is the vital component of the input to the application generator. Developers build an application by creating initial data and process models of the application, which they then refine into a more detailed design. From that design, an application is generated.

Lower CASE

Most Lower CASE application generators advocate a bottom-up approach to software development. The developer produces a prototype application and then iteratively amends it. Lower CASE tools provide highly integrated design facilities such as screen and report painters, pseudo-code editors, process debugging and testing and maintenance features (Figure 3). Although CASE tools provide many of these facilities, the integration of these tools, at the design and implementation level, may not be as tight or as wide ranging. Also, Lower CASE tools tend to be closely associated with a set of target hardware platforms, so an application can make better use of the target hardware's facilities.

Lower CASE tools make up for their lack of analysis tools by providing interfaces to a wide range of 'upper CASE' or analysis and design tools. They maintain their share of the application generator market by demonstrating their effectiveness at generating complex applications from design information.

Top-down or Bottom-up?

Choosing an application generator enforces a nasty choice on developers. Both the top-down approach of I-CASE and the bottom-up approach of Lower CASE are necessary. Having to choose between them is a compromise. The larger or more complex an application, the more assistance a developer requires in specifying that application. An I-CASE tool which supports techniques such as Entity-Relationship Diagrams or Data Flow Diagrams can be invaluable. However, in areas such as the development of a user interface, it is important to be able to develop the necessary software by iteratively amending an initial prototype so that the customer can view successive prototypes and recommend changes.

One way to get around having to make the 'nasty choice' between top-down and bottom-up is to use a CASE tool for the analysis and top level design of an application. A developer can then transfer the resulting specification to a Lower CASE tool to produce the detailed implementation.

This process is never so easy. Developers have to select the best tools to use from a large set of product types and products from many different vendors. Developers may have access to Upper CASE tools, 4GEs, 3GLs, project management tools and others. However, the interfaces between these tools are often rudimentary. On the whole, developers must work in disparate environments with tools which may not communicate.

Another trend, that of *interoperability*, is creating more disparate environments, rather than unifying environments. Interoperability means that developers can combine software tools, DBMSs and applications all from different vendors. It really means the ability to connect different products into a workable whole using standard interfaces.

The most successful example of interoperability is the use of databases with application generators. As a result of the acceptance of SQL as a standard data access language, developers can largely choose which database they wish their applications to use. Currently, most application generators support the use of SQL as a standard means of communicating with a database. Application generators should, therefore, be able to produce applications capable of using several databases. This type of integration is called DBMS interoperability. Products such as ACCELL, INGRES, UNIFACE and others, specialise in providing developers with such freedom. The acceptance of SQL as a standard has made databases a separate commodity. Developers select an application generator and then they have a choice of database to use for storing their generated application's data.

The pressure for 'disparate environments' is almost never-ending, as developers demand even more forms of interoperability and integration. For example, Current CASE tools, largely 'first generation', attempt to assist developers by providing an environment which supports the use of software development techniques and methodologies. The next generation of CASE tools is emerging, with greater emphasis on providing support for controlling the software development process and integrating with other development tools. For example, I-CASE tools such as Foundation, IEF, ADW/IEW and LBMS' Systems Engineer provide improved integration and control facilities.

Single Vendor Approach

The vendors of application generators are trying to increase the number of product types their tools can interface with. Some vendors believe that they can best get around the 'disparate environment' problem by developing all their own software tools. For example, a number of 4GE vendors provide their own Upper CASE tools. This 4GE/Upper CASE partnership is offered, for example, by Cognos with its PowerCASE and PowerHouse 4GL, Information Builders with its FACT CASE tool and FOCUS 4GL, McDonnell Douglas with its Prokit Workbench and PRO IV. Other single-vendor products include IEF, IEW and Corvision.

The advantages of this approach are that the integration of the Upper CASE and Lower CASE tools can be made seamless

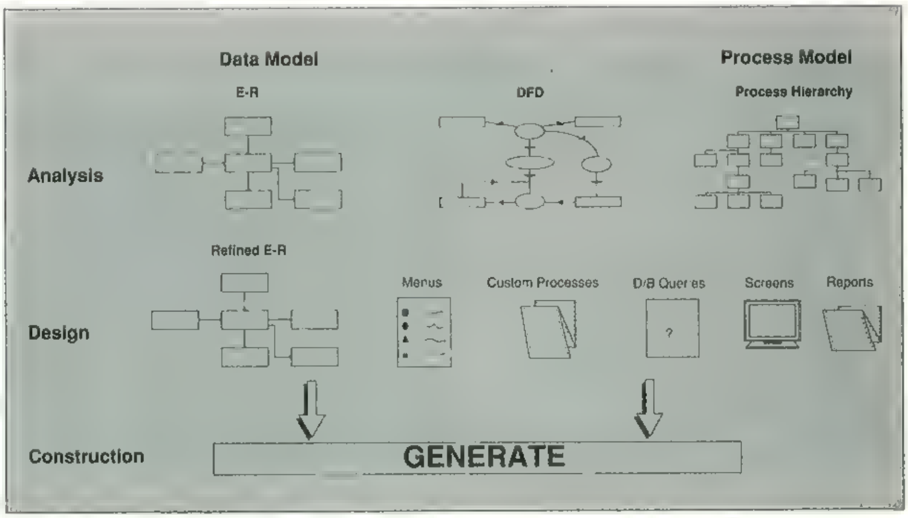


Figure 2 - A typical development strategy for I-CASE tools

and the tools can complement each other (ie the Upper CASE tool can take advantage of all the 4GE's facilities). The problem is that developers can feel locked in to a particular vendor if the tools do not interface with other development tools. It is especially important that an application generator can interface with other software tools. Despite the arguments of sales persons, no one application generator can provide automated support for all parts of the software development process. For example, few I-CASE tools provide project estimation tools.

All 'Interchange'

There is a new and exciting type of product which goes a long way to solving these classic dilemmas. Rather than adopting the above single vendor approach to a complete software development environment, developers could use 'interchange tools'. One example is Exchange from a company called Software One. Exchange can transfer the analysis and design information from an Upper CASE tool to a Lower CASE tool. It joins analysis tools, such as IEF, IEW, Foundation Design/1 and Excelerator, to Lower CASE tools, such as TELON and INGRES.

Using the interchange approach, developers are freed from any danger of being locked in to a particular vendor. However, the quality of the interface between development tools is dependent upon Software One's ability to produce adequate rule sets. The interfaces available are acceptable. For example, Exchange can generate SQL, screen layouts and skeleton processes for TELON from information in Upper CASE tools.

A problem that may occur when using Exchange, is deciding at what point to transfer development from the Upper CASE tool to the Lower CASE tool. Obviously, the more work a developer can do in the Upper CASE tool the better, since the Upper CASE tool will support a number of well defined methodologies. But some parts of a design are not transferable and developers will wish to avoid redoing work which could not be transferred.

Interfaces

Some vendors of 4GE and CASE tools have increased their products' ability to interface with other software tools by forming 'partnerships' with other software tools vendors. For example, a 4GE/DBMS vendor may negotiate with a CASE vendor to allow the contents of the CASE tool's data dictionary to be imported into the 4GE for further refinement. This benefits the CASE vendor since users of the CASE tool will have a choice of which Lower CASE tool to adopt. Similarly, developers using the 4GE may have a choice of Upper CASE tools on which to carry out their analysis and design. The 4GE can then import the results of that analysis and design. CASE tools, such as IEW and Prokit, have interfaces to a large number of Lower CASE tools. Also, Lower CASE tools, such as AD/Advantage, APS, Progress and Uniface, can import information from a range of Upper CASE tools. These interfaces, between different vendor's software tools, give developers more choice so they do not feel 'locked in' to a particular vendor.

There are disadvantages to this approach. The interface between tools may be primitive, each toolset may operate in a different environment with a different user interface and the tools may not always complement each other. Some current interfaces only permit data to go one way - from the Upper to the Lower CASE tool but not from Lower to Upper. With such an interface, developers could not replicate, in the Upper CASE tool, modifications made to a design, in the Lower CASE tool. So two designs would need to be maintained and suddenly, many of the maintenance benefits of application generators are lost.

Standards

Some of these integration problems can be resolved by standards. There is the Information Resource Dictionary System (IRDS) from ANSI and ISO, the Portable Common Tools Environment (PCTE) in the UNIX world and the CASE Data Interchange Format (CDIF). Hardware vendors are also proposing strategies to encourage integration and portability across their various machines and operating systems. Examples include IBM's Systems Application Architecture and Digital's Cohesion. However, these standards have yet to mature.

At present, developers wishing to acquire automated support for the software development process have a number of choices:

- Use an I-CASE tool and commit to a single vendor who may provide limited integration facilities with other vendors' tools.

- Use a Lower CASE tool in conjunction with paper-based analysis and design techniques in which the quality and maintainability of generated applications may suffer.

- Mix Upper CASE and Lower CASE tools using interfaces or relying on standards, both of which may not be ideal.

Currently, none of these approaches are perfect, but I-CASE may be the most appealing. By their very nature, I-CASE tools provide more substantial support for the development process than Lower CASE tools and the integration of their facilities should be less problematic than the integration of software tools from different vendors. However, I-CASE is not so dominant in the software development world so that a new challenge, in the form of repository technology, may yet emerge.

Repositories

The core of an application generator is its data dictionary or repository. This stores a model of an application's data, and data about the objects which make up the application. This information helps a developer to document and control software development through the stages of the software lifecycle. There is a great deal of interest at the moment in repositories, with IBM, DEC, ICL and others defining their own repository. When completed, the definitions of these repositories will be made widely available so software tool developers can enhance existing tools or develop new tools to interface with the repositories.

This new approach has been termed 'Component CASE' (C-CASE). It should allow

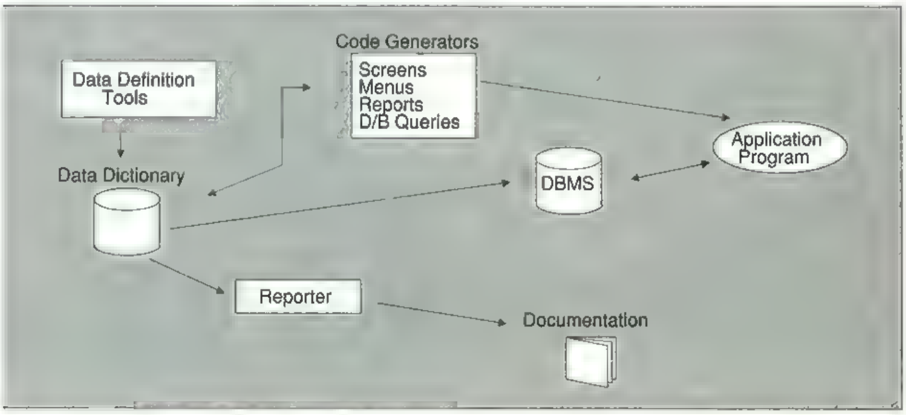


Figure 3 - A typical Fourth Generation Environment

If you have to get more out of your next C++ Compiler make sure there's a lot more in it

Environment friendly
TopSpeed's 10 Windowed, fully configurable environment makes for light work and light-speed

Make your code mean and lean
TopSpeed Smart-Linking eliminates unused code for faster tighter programming

Move into Windows
Support for MS Windows 3.0 and OS/2 Presentation Manager

TopSpeed – to the power of 4
C++, C, Pascal or Modula-2 can be compiled together using TopSpeed – the Multi-Language Programming Environment

Putting more into standards
100% AT&T 2.1 implementation of C++

Smash the DOS 640k barrier
Overlay Code & Data with TopSpeed

Search & destroy that Bug
Debug fast with TopSpeed's VID

Rogue Wave (tm) Class Library
In a class of its own, with over 50 C++ Math & Tools routines

Optimise your code, go TopSpeed
Fast & compact code every time

Project System

Hypertext Help
Context sensitive help system that you can tailor to meet the needs of a professional development team or a student programmer

Mouse support or Macro-keys
TopSpeed gives you more choice

Choosing your next compiler is no mean feat. Whether you're just putting a toe in the water or about to embark on a major development project with C++, you'll need to know that the compiler you choose gives you the productive edge.

***TopSpeed* C++ packs more in**

TopSpeed C++ packs more in, whether you are developing for DOS, Windows 3 or OS/2. Develop mean-and-lean code with TopSpeed's super-optimizing code generator, Smart-Linking and Project system. Then, add the most extensive C++ class library available and you too can now harness the real power of C++.

***TopSpeed* Putting more into standards**

TopSpeed C++ not only complies with the industry standards offering 100% conformance to AT&T V2.1, but also packs in a range of extensions to take your programming into hyperspace!

***TopSpeed* just the environment for C++**

To make the most out of the C++ and the power of OOP you'll need the development environment that packs the most in. TopSpeed's unique Project system frees you to be creative whilst it manages file dependencies, compilation, assembly, linking and much more.

***TopSpeed* C++ in a family of thoroughbreds**

C++ is just one of a family of four compilers that integrate seamlessly into the TopSpeed Multi-language Development Environment. From $99 per module you can custom-build a development system that takes your programming TopSpeed!

For more information call JPI on the number below.

EUROPE
+44 (0) 234 267500

CIRCLE NO. 848

JPI
Jensen & Partners International

USA
(415) 967 3200

developers to pick and mix different vendors' tools using the same repository throughout the development process. It could be a solution to the 'disparate environments' problem.

The problem with the C-CASE strategy is that none of the generic repositories are currently available in a form which supports the bulk of the activities needed for modern software development. Defining such generic repositories is not a trivial task, since it must be able to store data from a wide range of software tools, eg existing analysis and design tools, project management tools and many others. Until these repositories are available, I-CASE vendors may prosper. Developers and vendors cannot wait indefinitely.

The alternative to using I-CASE or waiting for C-CASE, is to use an interchange tool like Software One's Exchange. Exchange allows Upper CASE tools to export the information in their repositories to Lower CASE tools, for the generation of an application. And it gives the appearance of C-CASE.

Until the arrival and acceptance of proven generic repository technology, it will be I-CASE tools and interchange tools which will be the first port of call for most developers. When generic repositories do appear, developers may still prefer I-CASE to C-CASE since most developers would rather use one or two vendors' tools than try to combine software tools from many vendors.

Increased Automation

But what of the 4GEs and other Lower CASE tools? Current application generators do not automate a large percentage of the development process. Developers spend a significant part of their time using screen painters, report writers, pseudo-code (or 4GL) editors and other low-level design tools. Future generator products should reduce the need for these low-level tools by generating applications from analysis and high-level design information. Some application generators already support the generation of database schemas and data access operations from Entity Relationship diagrams. It may also be possible to generate more of an application's design using the deliverables from other techniques. For example, Data Flow Diagrams (DFDs) could be used to help define a user interface since in a DFD, user interface is required between external entities and processes. Thus, developers would be able to specify an application using high-level diagrams and the bulk of the application could be generated automatically.

As a result of the increased automation facilities in CASE tools the need for separate Lower CASE tools, (including 4GEs) would be reduced. The application generator would be able to infer the bulk of an application's design from its analysis. It may even result in COBOL, so often presumed dead, outliving its 4GL adversary since the use of 4GLs will recede (or be consumed by I-CASE) but 3GLs may remain as the 'machine code' of I-CASE.

EXE

Patrick McParland is a consultant with the Institute of Software Engineering. He has just completed an in-depth study of application generators entitled Application Generation: Automated Software Development using 4GEs and CASE. *This report is available from the Institute of Software Engineering (0232 738507) at a cost of £325. This article was generated manually by the use of keystrokes.*

Windows3.0 ToolBox

Power up your Windows apps!

Over 200 functions!

With over 200 functions, Windows Toolbox is designed to enhance, simplify and dramatically speed up the development of Microsoft Windows™ applications.

Flatten the learning curve.

The growing popularity of graphical user interfaces and the release of Windows 3.0 means more and more programmers will be entering the Windows environment, but the learning curve associated with Windows development is steep. Windows ToolBox short circuits this development cycle, providing the higher level functionality that Windows applications require.

Seven new classes!

Time and *date* classes are accompanied by a variety of alternative display styles (12 and 24 hour time formats and the various international date strings). Related functions convert time and date functions to and from strings.

For costing and financial applications, the *money* style adds refined capabilities over and above those of float data types.

Alistbox class is supported with numerous functions to ease user selections and lock out costly data entry errors.

The *edit* class makes manipulation of text a breeze, while the *integer* class expands upon the primitives supplied in the Microsoft SDK.

To display processing, ToolBox adds a *gauge* class with thermometer style readouts.

£249.00

PCX
Effects & Text

IMAGE Solutions!

VBase Colour Database £249.95

Turn your dBase, Clipper, C, Basic, Pascal, application into a powerful 256 - COLOUR DATABASE with little programming skills. Includes full printer support ! Most any application can be converted in just a few hours.

PCX Programmers Toolkit £165.00

Includes over 75 routines to display, save, scale, and print PCX bitmapped graphics from almost any programme. Full use of virtual buffers and EMS 4.0.

PCX Effects £85.00

Add spectacular special effects (F/X) to your program. Wipe, split, crush, slide, sand, drip, diagonal, spiral, random, or explode your graphics. Requires PCX Toolkit.

PCX Text £149.00

Display blazing bitmapped text in any graphics mode as simple as text mode. You can display your text information on top of graphics.

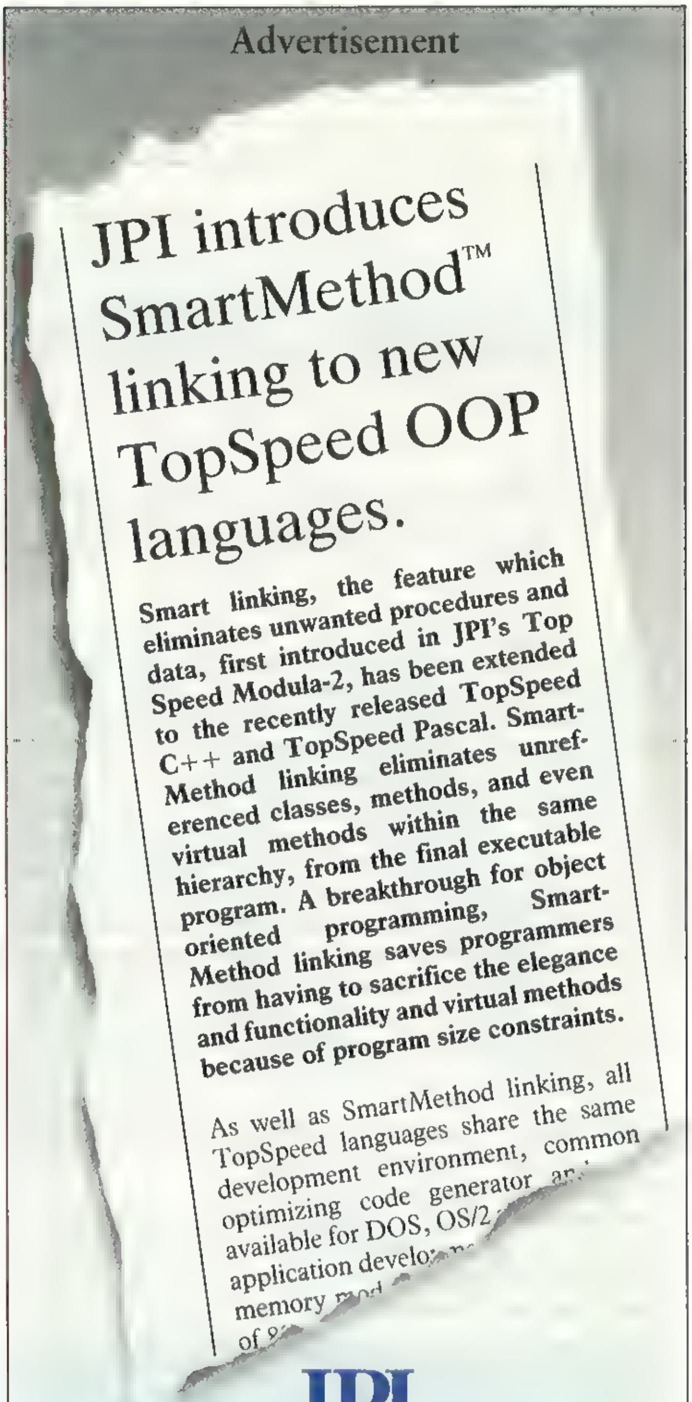
GX Graphics £179.95

A complete graphics library supporting all graphics primitives. Use the GX Graphics Toolkit instead of the BGI or MS libraries and make your program faster, smaller and more portable across compilers - while supporting more video modes.

**Pinna Electronics Ltd., APL Centre,
Stevenston, Ayrshire KA20 3LR
Tel: (0294) 605296 or Fax: (0294) 68286**

CIRCLE NO. 851

Advertisement



JPI introduces SmartMethod™ linking to new TopSpeed OOP languages.

Smart linking, the feature which eliminates unwanted procedures and data, first introduced in JPI's Top Speed Modula-2, has been extended to the recently released TopSpeed C++ and TopSpeed Pascal. SmartMethod linking eliminates unreferenced classes, methods, and even virtual methods within the same hierarchy, from the final executable program. A breakthrough for object oriented programming, SmartMethod linking saves programmers from having to sacrifice the elegance and functionality and virtual methods because of program size constraints.

As well as SmartMethod linking, all TopSpeed languages share the same development environment, common optimizing code generator an available for DOS, OS/2 application develo memory mod of 9

JPI
Jensen & Partners International

3 The Mansards, Tavistock Street, Bedford MK40 2RX
Telephone: 0234 267500 Fax: 0234 217094

See JPI's new SmartMethod Linking in action at
SOFTWARE TOOLS '91
Wembley 11-13 June.

Please send me more information on TopSpeed SmartMethod linking. □
I program in the following languages:
C++ □ Pascal □ C □ Modula-2 □

Name: ________________
Company: ________________
Position: ________________
Address: ________________
____
________________ Post code: ________
Telephone: ________________

CIRCLE NO. 852

Doing CASE on Windows 3.0

Can stalwart programmers, with overworked backs of fag-packets, be converted to CASE tools? Tracy-Anne Ormrod reviews a candidate Marlboro alternative: Systems Engineer.

LBMS was founded in 1977 as a consultancy company with particular expertise in the database design field. After developing SSADM in conjunction with the CCTA the company marketed its own method LSDM. The first CASE tool LBMS produced, Automate, marked a new direction for the company, whose subsequent strategy has been based around the production of CASE tools with the training, methods and consultancy activities as peripheral. LBMS's Systems Engineer was originally destined for Windows 386. A prototype was commissioned and a completed product emerged onto the market in Spring 1990 to run under Windows 3.0.

Systems Engineer (SE) is a multi-user Windows 3.0 CASE tool available in two different method flavours; an enhanced LSDM called Systems Engineering and SSADM V4. The tool uses SQLBase, Gupta Technology's powerful client-server relational database management system. By using SQLBase as the development platform LBMS has chosen to build their tool with an Open Architecture.

For the purpose of this review I used the Systems Engineering version as a single user, although I will be covering multi-user features. The hardware required to run the tool is what you would expect of a Windows 3.0 application: for a single user system ideally 80386 25 MHz or above with a minimum 4 MB of extended memory (preferably 6 MB) and about 30-40 MB of hard disk space to accommodate the product and a reasonable size of project. The review machine was a Research Machines Nimbus 386/25 MHz with 4 MB of RAM and 100 MB hard disk.

Installation and setting up was straight forward enough for a single user system. However, LBMS recommends that its engineers install any multi-user system, which conjures up all sorts of support nightmares in the imagination.

On-line help is available on both tool and method which gives novice users of this method a boost. The help facility is not context sensitive and occasionally a little short on explanatory text. A tutorial is supplied in addition to the main help facility, to enable users to get to grips with the tool quickly. The product arrives with a set of three manuals, a system manager's guide, a guide to the use of Windows and the menu functions, and an application guide, describing each facility within SE, the use of the Design Editor and techniques. Guides to the Systems Engineering method are purchased separately. I found the writing style of the manual bland and lacking in enthusiasm. Perhaps LBMS is relying a little too much on SE's ease of use and help facilities. The manuals do not answer every question posed by the user and require, in my opinion, 'padding out'.

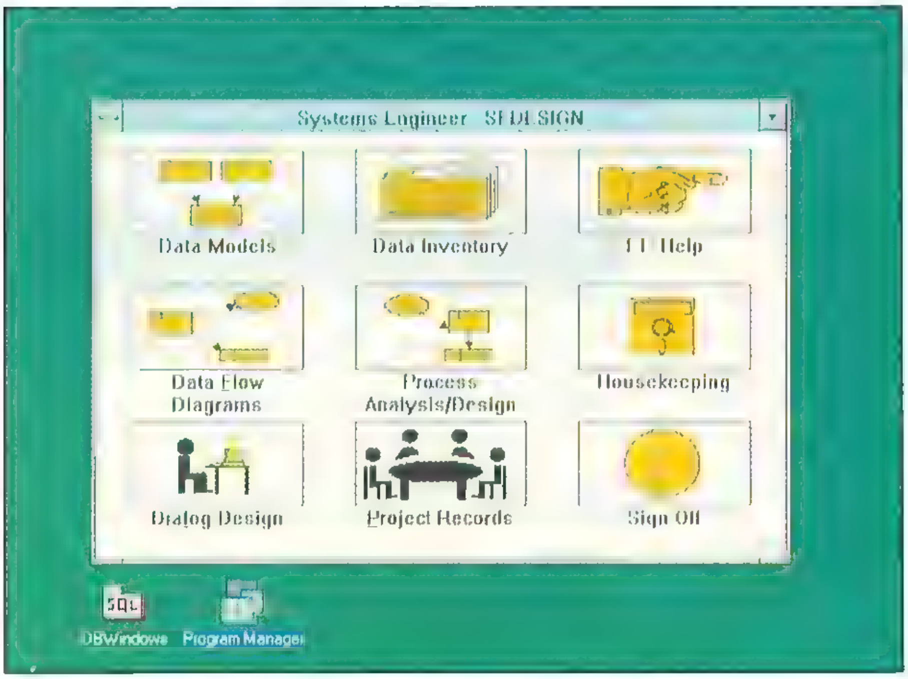


Figure 1 - System Engineer's Main Menu

ADVANCED BUSINESS GRAPHICS

The Facilities

The facilities are arranged around a main menu of non-standard icons accessed by pointing a cross mouse cursor and double-clicking in the usual manner. Facilities offered are: Data Models, Data flow diagrams, Dialog Design, Data Inventory, Functional Analysis, Project Records, Help, Housekeeping and Sign Off. Selecting a facility either invokes another icon menu or produces a dialog box asking for further information. Security is based around each user having a sign on name and password, entered before access to the facilities menu is granted. Product security is in the form of the ubiquitous dongle attached to the printer port.

As an analyst and developer first and a journalist second, I set out to use the tool as I would when developing a system: an iterative approach with changes being made,

DOS/16M
DOS/4G

PROGRAMS TOO LARGE FOR DOS 640KB?
PORTING FROM UNIX TO DOS?
PROGRAM FUNCTIONALITY LIMITED DUE TO DOS MEMORY CONSTRAINTS?

DOS/16M™

DOS/16M is Rational Systems' 16-bit DOS Extender that allows your programs to directly address up to 16 MEGABYTES of extended memory on any 80286/386/486-based PC.

DOS/16M's library extends your compiler's run-time library so it can allocate and use extended memory with little or no change to existing code. Programs built using ***DOS/16M*** can run in as little as 18 Kilobytes of DOS memory.

DOS/16M supports most popular C, C++, FORTRAN, Pascal, and Modula-2 compilers, so you won't have to change the tools you're using.

DOS/4G™

DOS/4G is Rational Systems' 32-bit DOS Extender. By utilizing a 32-bit flat memory model ***DOS/4G*** makes it easy to build high performance applications or port existing UNIX applications to run in protected mode under DOS.

DOS/4G offers you the same low DOS footprint as *DOS/16M* while giving you access to up to 4 GIGABYTES of extended memory on any 80386 or 80486-based PC.

DOS/4G supports the Metaware C, Watcom C and Zortech C++ compilers. Support for additional compilers will be announced with upcoming releases.

Both ***DOS/16M*** and ***DOS/4G*** support the DPMI and VCPI standards, giving your program compatibility with other extended memory programs, as well as Microsoft Windows 3.0/3.1.

Included with ***DOS/16M*** and ***DOS/4G*** is a protected mode source level debugger with a variety of performance measurement and analysis capabilities designed to take full advantage of protected mode's run-time checking for faster, easier debugging.

With ***DOS/16M*** and ***DOS/4G*** you receive expert technical support directly from the actual development team -- not a separate support staff.

Join companies such as IBM, Lotus, DEC, Ashton-Tate, Computer Associates, SITKA/TOPS, DataEase, Informix, (plus hundreds of others), in using ***DOS/16M*** and ***DOS/4G*** to develop your 16 and 32-bit protected mode DOS programs.

CONTACT RATIONAL SYSTEMS TODAY FOR YOUR *DOS/16M* AND *DOS/4G* INFORMATION PACKAGE.

Tel: 1 508-653-6006
FAX: 1 508-655-2753

Rational Systems, Inc.
220 N. Main St.,
Natick, Massachusetts
01760 U.S.A.

Trademarks are acknowledged to Ashton-Tate, Computer Associates, DataEase International, Digital Equipment Corporation, International Business Machines, Informix, Lotus Development Corporation, Microsoft Corporation, Rational Systems, Inc., and Sitka.

CIRCLE NO. 853

different diagrams being added and used when needed. I started with the Project Records facility to list problems and requirements for the system. This facility allows the user to define the problems of the existing system, the requirements for a new system, and select a priority for the particular problem or requirement. These can then be cross-referenced with any other Design Object. For example, to link a requirement to a particular process on a data flow diagram or a screen in the ultimate system. Solutions to problem/requirements can be maintained here, with a solution being associated with a particular problem or requirement. In addition to problems and solutions, Project Records include a General Forms application with several uses: adding additional text to other Design Objects, recording cost/benefit analyses, and tracking change requests from other analysts on the team.

SE has several diagramming techniques available to the analyst or designer which cover what is required by the Systems Engineering method. A useful addition, also within the Project Records facility, is the General Pictures Application giving three extra picture types to use. A Network Entity Life History picture concerned with the time orderings of the transactions, also investigates system logic and can provide a basis for error conditions and handling. The second diagram is used to create a Batch Suite picture, which can help show the programs, their sequence and the flow of data between programs, databases and transaction files. The third diagram, System Structure, is a general purpose diagram which can be used to depict a high-level graphical view of the system. It has various friendly objects such as a Floppy disk, Report, Lorry, Telephone etc which enable the diagram to be a useful communication tool between analyst and user.

Data Flow Diagrams

After defining requirements for the system and producing a high-level diagram such as the System Structure picture, one would normally continue with data flow diagrams. When constructing data flow diagrams, my method of working varies between top-down or bottom-up, depending on the system that is being built.

This is where my first gripe with SE came. Because of the method behind SE and its control over the product, it is not possible to construct bottom-up. Data flow diagrams in the tool are based around a 'set' or collection of DFDs. A specific project can have a number of DFD sets. After naming the set and saving the description you have to select Picture from the DFD set menu to access the drawing area. This then becomes the top level diagram. From here on, the only way to progress is downwards.

Because of the method behind SE and its control over the product, it is not possible to construct bottom-up

The drawing area is divided up into a grid for placing objects on the area, an icon menu, and standard Windows features such as a menu and title bar across the top and scroll bars. Populating the drawing area couldn't be easier: the mouse is pointed to the appropriate icon to activate it and then dragged into position. Once a position has been selected a dialog box appears for details to be entered. A maximum of 32 characters has been allowed for names, which I personally found to be constraining. It may sound a large label area but process names do not have to be verbose to reach the limit. I found myself shortening names to fit in the allowed label, therefore reducing the clarity. Continuing at a detailed level, moving and constructing the diagram is a much more pleasant experience compared to the so-called first generation CASE tools such as Automate Plus, Excelerator or IEW.

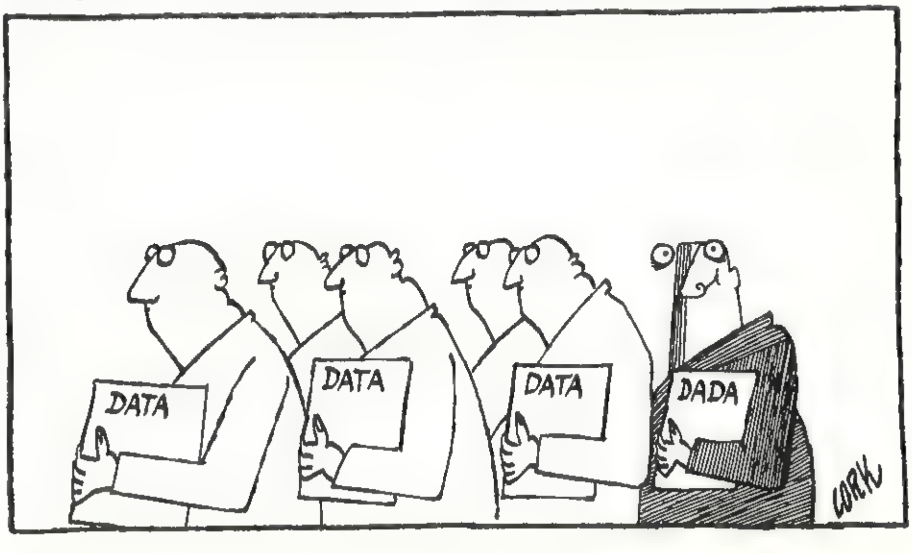
Creating levels within the DFD set is achieved by choosing the process to be decomposed and selecting the View Decomposition command. The top-level diagram is saved and all data flows connected to the process are carried down to the lower level diagram. The method behind SE dictates that the System Boundary box cannot be removed, so all external entities have to be displayed on each relevant lower level diagram. I find that this often clutters up the diagram unnecessarily, making the diagram difficult for both analyst and user to understand.

Validation of diagrams takes place on a per diagram basis. The series of checks identify objects not connected to any others, processes and data stores that have only incoming data flows and vice-versa. When lower level diagrams are validated, the checks made are more extensive; the parent and child picture are compared to each other with all processes, data stores and external entities on the parent diagram but not on the child diagram being flagged. The diagram checker also looks for objects to be connected in the same way on both parent and child,diagram.

The tool is comprehensive in its validation, checking and reporting capabilities. Reports are available on all the objects in the current DFD set, together with discrepancy reporting on data stores, which compares the Data Items associated with incoming or outgoing data flows and the data stores itself. Data stores which contain different data items from the data items on a data flow will be reported on.

Data Modelling

SE has a comprehensive, but not extensive, set of symbols for modelling the entities within your system. The data model must be assigned a type which can be either one of the following: Current Data Model, Required Data Model, Third Normal Form Structure Model, Composite Logical Data Model or a type of your own.

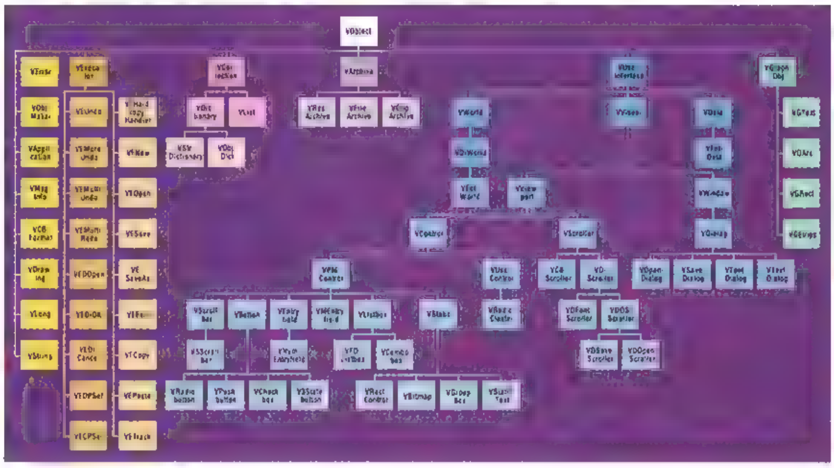


Effectivity in developing MS Windows or OS/2 Presentation Manager programs

If you are building sophisticated C++ applications for MS Windows or OS/2 PM, you could benefit from

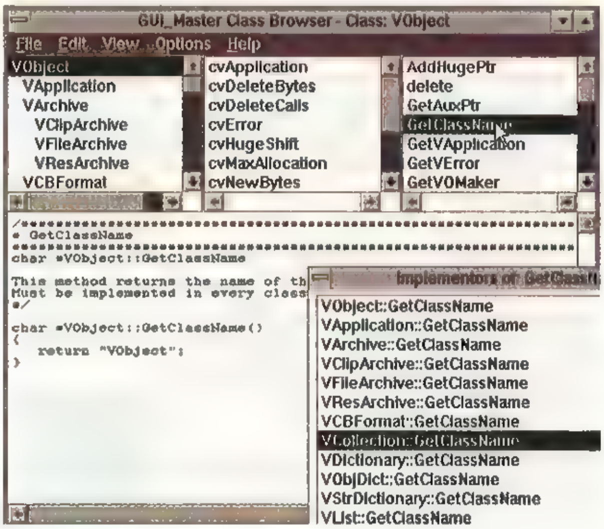
- A toolbox with standardized Graphical User Interface (GUI) parts
- An Interface Builder which constructs the standard parts of your program for you
- A Browser so that you can examine and edit the numerous pieces of code of your application

We introduce our *GUI_Master (CLASS-TREE for C++)* which we originally developed for our own use to lessen the burden of writing programs so that one can focus on the application parts and not on the GUI part.



You don't really want to invest your valuable time in reinventing the GUI-wheel?

- We don't, so we included over 85 classes containing a wide range of GUI building blocks. Anything that could be accomplished through "normal" programming can be done if you use our product. It's just a lot easier
- Many of the things you might not include in your application because they're too complex or time-consuming to program, are easy to make with *GUI_Master (CLASS-TREE for C++)*
- We provide the Interface Builder, with which you specify all properties of the visual objects of your program. The interface builder then generates the necessary source code, the resource specifications and even the make file
- The Browser enables you to examine your own code and code made by others, and to understand the class structure of that code. From the Browser you can launch the editor of your choice to modify a C++ program


GUI_Master Class Browser - Class: VObject
File
Edit
View
Options
Help
Implementors of GetClassN

VObject
VApplication
VArchive
VClipArchive
VFileArchive
VResArchive
VCBFormat

cvApplication
cvDeleteBytes
cvDeleteCalls
cvError
cvHugeShift
cvMaxAllocation
cvNewBytes

AddHugePtr
delete
GetAuxPtr
GetClassName
GetVApplication
GetVError
GetVOMaker

```
/*************************************************************************
* GetClassName
**************************************************************************
char *VObject::GetClassName

This method returns the name of th
Must be implemented in every class
*/

char *VObject::GetClassName()
{
   return "VObject";
}
```

VObject::GetClassName
VApplication::GetClassName
VArchive::GetClassName
VClipArchive::GetClassName
VFileArchive::GetClassName
VResArchive::GetClassName
VCBFormat::GetClassName
VCollection::GetClassName
VDictionary::GetClassName
VObjDict::GetClassName
VStrDictionary::GetClassName
VList::GetClassName

You may want detailed reference documentation

- We supply nearly 1200 pages of detailed class descriptions, method reference etc. An index is included, of course

You may need a cookbook and example programs

- We provide 7 functional example programs and a cookbook explaining the "how to's"

From our experience developers want to use products from someone who knows what he is talking about

- Vleermuis Software Research (VSR) is an independent R&D organization, which, over the past 3 years, spent over 40 person-years on OO development on a broad spectrum of commercial workstations

To use our *GUI_Master (CLASS-TREE for C++)* you must have

- A C++2.x compiler, as we do not supply a compiler with our product. We suggest using the Zortech compiler, but we have also tested most of the others. Write for details

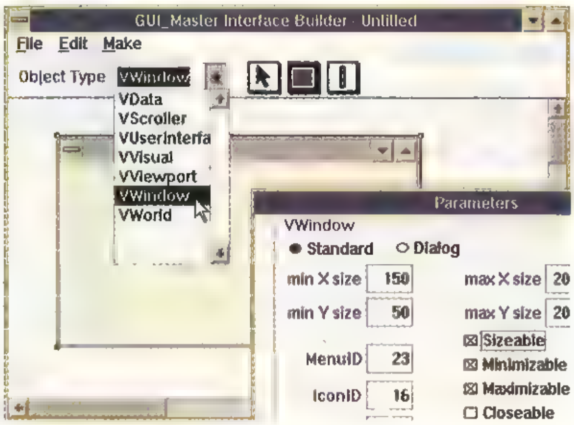
We try to keep our prices reasonable

$ 495.= **OS/2 PM version**
$ 545.= **MS WINDOWS 3.0 version**
Prices include airmail delivery (may take three weeks)
Add $ 50.= for courier service
General availability in January 1991 for the OS/2 version
MS Windows version in May 1991

Write or fax your order details to:

Vleermuis Software Research bv
P.O. Box 2584
3500 GN Utrecht
The Netherlands
Fax: Intl. + 31 30 310426

Use Mastercard (number and expiration date) or certified cheque



Object Type VWindow

VData
VScroller
VUserInterfa
VVisual
VViewport
VWindow
VWorld

VWindow
● Standard ○ Dialog
min X size 150 max X size 20
min Y size 50 max Y size 20
☒ Sizeable
MenuID 23 ☒ Minimizable
IconID 16 ☒ Maximizable
☐ Closeable

CIRCLE NO. 854

The Data Model can be split into one or more subsets. The Data Model Subset, as its name suggests, contains a subset of the entities on the Data Model picture. The entities remain part of the original Data Model but can be laid out differently. The advantage of having subsets of a Data Model becomes clear when you have a large Data Model which is maintained or needs to be maintained by a number of developers. If the model is broken down into subsets each subset can be accessed by one person for editing purposes. Where the model is maintained as a whole, only one person can edit - all others being restricted to browse only.

The Data Modelling techniques within SE allow for the use of entity, operational master, Access Entry Points, Direct Relationships, Optional relationships, exclusivity details and exclusivity master notation. Relationships are represented by a line with an arrow on the 'many' end linking the two entities. Many-to-many relationships between two entities can only be expressed by introducing a linking entity. Relationships can be named but the choice is left to the developer, the only exception being when there is more than one relationship between two entities, in which case one of the relationships must be named. The volume and volatility of the relationships can be recorded to work out capacity planning.

The entities and relationships are placed on the drawing screen in the same point-and-click manner as available in the DFD modelling. The entities are associated directly to Data Items which can be accessed from a selector list.

There is no explicit validation, although the tool does prevent you from what it perceives to be illegal connections or symbols and inconsistencies. Reports are available which show the current content of the data model and using the associations built up during work on the model it is possible to ensure that each data store on the DFD is associated with at least one entity.

Prototyping

It is always important to introduce the user to the 'look and feel' of the system as early on in the development life-cycle as possible. Having a CASE tool that incorporates a screen design facility is a must. SE has a Screen Map editor together with Menu Control Structure pictures. These features together with a pseudocode editor make up the prototyping facilities available to the developer.

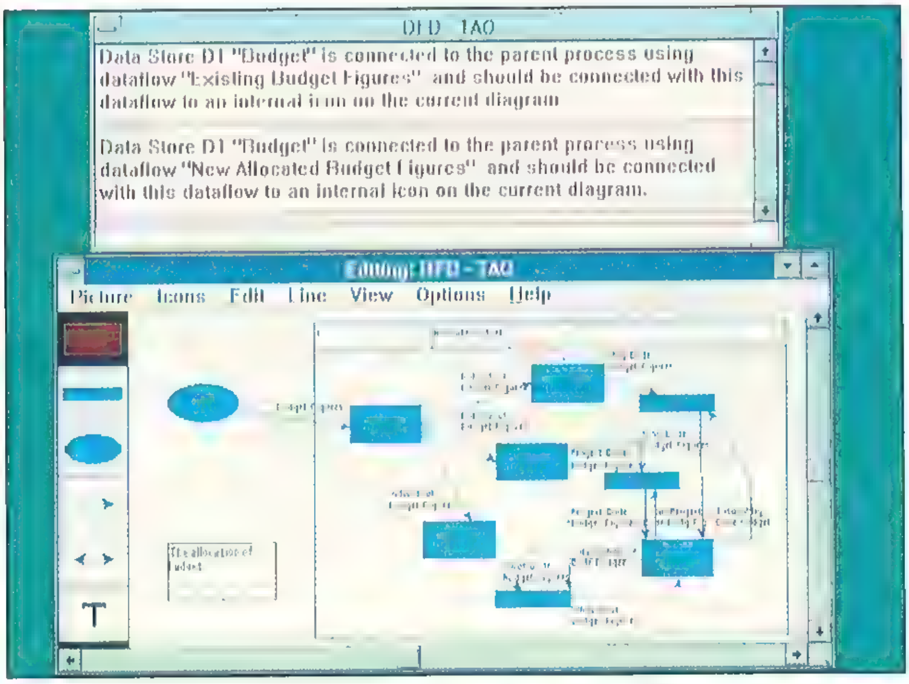
By using SQLBase as the development platform, LBMS has chosen to build its tool with an Open Architecture

The Screen Applications area has two major components: the environment, where the Data Items displayed on the screen are defined with five different types of field, and the screen layout where the positions of the Data Items and any other fields are specified. The Data Items must be defined first before entering the Screen Map editor and to tie in with your complete design the Data Items used should already exist on the database. The screen design area allows for difference in colours, highlighting and placement of labels. I understand that the Screen editor is due for enhancement and it is hoped that a GUI screen design editor will be available within a year. This will be a welcome addition to the character-based screens that are all that can be designed at the moment.

The Menu Control Structures let the developer plan and describe how the system will interact with the user. By building up a set of pictures it is possible to navigate through the proposed system using the Prototyper Application and the Screens that have been designed. The three types of Structure - Menu Control, Global Definition and Transaction Dialog Structure - become the navigation tools around the system. The Prototyper Application is used in conjunction with the Control Structures and Screens to demonstrate the flow of control through the system's menu structure. The Prototyper has three different modes: Menu level, Prototyping and Prototyping with screens. Full prototyping with screens gives a realistic walk through of the menu and screens and the keys used to navigate around the system.

While I can see some developers using this area of the tool, those with comprehensive libraries and languages that lend themselves to quick 'knock-ups' of screens will find this alternative a waste of valuable time. The tool currently supports IBM IMS DC COBOL DB2 via the Application Engineer tool and Oracle, DB2 and Informix via interfaces. I would like to see this part of the tool together with the Pseudocode Editor offering more generation of code and in more languages than are currently supported.

The Process Design Facility is used to produce Module Decomposition Charts and pseudocode. A Module is defined in SE as a logical unit of processing which can call other modules. Using this definition, a hier-



Data Store D1 "Budget" is connected to the parent process using dataflow "Existing Budget Figures" and should be connected with this dataflow to an internal icon on the current diagram

Data Store D1 "Budget" is connected to the parent process using dataflow "New Allocated Budget Figures" and should be connected with this dataflow to an internal icon on the current diagram.

Figure 2 - Data Flow Diagram together with validation checking

ADVANCED BUSINESS GRAPHICS

C Breeze.

Get set for smooth sailing. Because Microsoft® C Professional Development System version 6.0 is not only designed to be fast and powerful, but also easy to control.

Thanks in large part to Microsoft's unique Programmer's WorkBench.

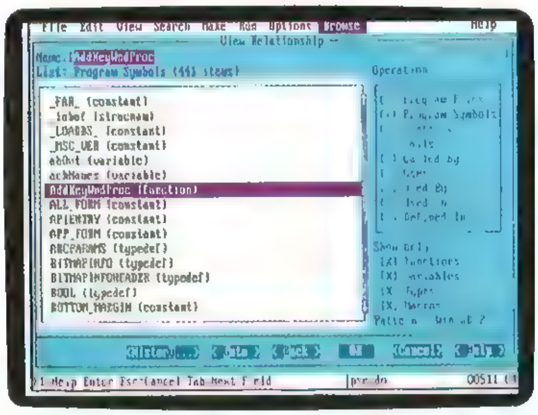
It integrates the debugger, editor and compiler, allowing them and all the other powerful programming tools to share data.

Tools such as our Source Browser, that taps directly into the project database. This feature will give you instant access to the relationships between functions, macros, types and variables. It'll even give you a full call tree that literally draws you a map.

With our CodeView® Debugger, you'll save a lot of time. This powerful tool not only debugs any size DOS* or OS/2 application program, on any 286 or 386™ machine, but also lets you explore nested structures and arrays.

The programmer's WorkBench even has an open architecture that is supported by many third party tools.

Our online and hardcopy documentation are designed to complement each other. Together they provide you with an intelligent mix of professional guidance.



Name: AddKeyWndProc
List: Program Symbols (441 items)

```
_FAR_ (constant)
_iobuf (strucnam)
_LOADDS_ (constant)
_MSC_VER (constant)
abOut (variable)
achNames (variable)
AddKeyWndProc (function)
ALL_FORM (constant)
APIENTRY (constant)
APP_FORM (constant)
ARCPARAMS (typedef)
BITMAPINFO (typedef)
BITMAPINFOHEADER (typedef)
BOOL (typedef)
BOTTOM_MARGIN (constant)
```

( ) Called by
[X] Functions
[X] Variables
[X] Types
[X] Macros

F1=Help Enter Esc=Cancel Tab=Next Field

Explore the relationships between functions, variables, types, and macros using information created by the compiler and stored in the project database.

You'll find the C Advisor always online and ready to help. It is a hypertext-based reference system complete with sample coding solutions you can copy and paste directly into your program.

You will also find the hardcopy documentation offers many useful tips and techniques for advanced C programming.

So whether you're developing applications for the MS-DOS® Microsoft Windows™ version 3.0 or MS® OS/2 Presentation Manager systems, our Programmer's WorkBench will allow you to write the fastest code around in the shortest amount of time possible.

For a free white paper with more details, just call Microsoft on (0734) 500741.

Then get your hands on Microsoft C version 6.0. It'll blow you away.

Microsoft®

Microsoft Languages and Development Kits may be purchased from your usual dealer or direct from the following Microsoft Language Resellers: Grey Matter, 0364 53499. Software Construction Company, 0763 244114. System Science, 071 833 1022 QA Training, 0285 655888. *This requires 550K of available extended memory. ©1991 Microsoft Corporation. All rights reserved. Microsoft, MS, MS-DOS, the Microsoft logo and CodeView are registered trademarks and Windows is a trademark of Microsoft Corporation. 386 is a trademark of Intel Corporation. 386-Max is a trademark of Qualitas, Inc.

Objects At Your Fingertips.

Now, if you want to develop applications for Windows 3.0, there's a fast and easier way to do it with the premier object-oriented programming language. Smalltalk/V.®

With Smalltalk/V Windows, you can explore, prototype, build finished applications and ship them runtime free.

You can tap into applications using DDE so effortlessly you don't have to be a Windows expert to do it.

And with one of the world's most comprehensive class libraries, you can choose our objects or easily build your own.

But whatever you develop, it will be portable between the Windows, OS/2 and Mac versions of Smalltalk/V.

With so much at their fingertips, more people are solving more problems with Smalltalk/V than any other object-oriented programming system.

At only £330 + VAT and no runtime charges, you can solve them, too.

Just call us at 071-436 9481. And see why programming Windows has never been easier.

Smalltalk/V Windows

COCKING & DRURY (SOFTWARE) LTD

180 Tottenham Court Road, London W1P 9LE Phone 071-436 9481 FAX 071-436 0524

Smalltalk/V is a registered trademark of Digitalk, Inc. Other product names are trademarks or registered trademarks of their respective holders

CIRCLE NO. 846

archical chart can be built up of the modules. The chart itself does not show any processing logic on it, the pseudocode editor is used to show that. SE supports a pseudocode developed by LBMS called System Design Language. SDL lets you have a more precise specification of the processing structure than using Structured English or another derivative. The pseudocode uses largely the constructs that most programmers are very familiar with, for example; Case.., If..Else, For.., While.. I understand that the constructs used can be changed, which would allow for different languages or company standards. The pseudocode can contain calls to other modules and references to other design objects. All these are maintained automatically to ensure consistency. One nice touch is the colour set-up, giving complete choice over the colour of the module names, design object names, constructs and database access statements. I would like to use this part of the tool for transference into the target language or in the future perhaps as the basis of maintaining an existing system that has used SE as the original specification and development tool.

> ***At a price of £8,000 per user I expected to use a market leader CASE tool and I reviewed one***

Add On Tools

Systems Engineer comprises only part of the development life-cycle but together with other LBMS tools and the interfaces available the whole development life-cycle can be covered. The buzz word 'I-CASE' has been used to lure large development departments into purchasing a range of tools that make a patchwork quilt sown loosely together. SE is not an integrated CASE environment. LBMS claims to have an integrated set of tools that fit closely with SE making transfer of design information to other environments possible. One of these tools is SE/Open which is a universal interface providing facilities to transfer system designs created using SE to 4GLs, code generators, DBMS, or data dictionaries. Where you require a transformation that is not already in the library, LBMS will assess the feasibility of the interface and quote a fee for doing the work. Alternatively you can create your own interfaces using the SE/Open language and compiler available. One other tool worth noting, particularly for team leaders or project managers, is Workplace. Priced at £1,500 per user, it allows other desk-top tools to be integrated to SE. Workplace integrates SE with all other Windows tools that use the DDE facility. The tools can then be configured to access design information used in the SE design database without the need to know SQL. Other components of Workplace address project estimation using estimating parameters tailored precisely to your project and Method Online which gives immediate access to hypertext documentation on the method without the constraints of a manual.

Future directions for both SE and LBMS include an object-oriented approach or method, a more generic method (perhaps with some degree of user configurability) and the realisation that the PC developer has been largely ignored in the CASE explosion. All these ideas are largely on the drawing board and will take place dependent on market requirements.

Conclusion

At a price of £8,000 per user, I expected to use a market leader CASE tool and I reviewed one. I was impressed by SE's ease of use, its completeness and the thought that has obviously gone into the tool. There are still a few gaps to be filled but I feel LBMS should have these filled quickly. I feel that the tool will encourage more developers to examine the way in which they develop systems and encourage stalwart programmers still using the back-of-a-fag packet analysis method to enter into the CASE tool arena. Everything considered, the tool will appeal to a much larger section of development personnel, not just analysts, but programmers and maintenance personnel as well because it is a product that every developer will want on his desk. EXE

Tracy-Anne Ormrod is a self-confessed CASE tool and methods freak. Previously employed at Excelerator, she is now a director of Applications Technology (0491-35187), a small consultancy company specialising in systems development, CASE tool advice and training.

LBMS is on 071 636 4213.

Bearing the Standard

Programming standards have an important part to play in the production of high quality software, yet we are often unwilling to invest time and effort in their production.

Computer lore is full of stories about the early days of the industry. In the old days, the programming community was made up of enthusiastic academics who laboured into the night, pushing the limits of the new technology and making the rules up as they went along.

These days, the stakes are higher. Faced with ever more complex problems, the industry is looking to engineering disciplines for inspiration and such seat-of-the-pants programming is no longer acceptable. In a business where team-work is the norm, a coherent set of ground rules is a must for any non-trivial development project.

Mission Incomprehensible

There are many good reasons for having a programming standard. Source code is a program's most detailed, and sometimes only, technical documentation. Its quality can have a huge impact on the effort needed to make changes, particularly when the original team members have gone their separate ways. It goes without saying that delivering sloppy or incomprehensible source code to a client will leave a bad impression and can destroy any chance of future work.

The case for readable source begins when the first line of code is cut. Programmers need to read each other's code throughout the development cycle, and eventually someone has to put all the bits together and make them work. Of course, they never work first time, and effort spent navigating someone else's code just gets in the way of the real work.

Another issue which demands rigorous ground rules is portability. Even if portability is not a requirement, it is good practice to take it into account anyway, since the scope of the project may change. In any case, there will be constraints unique to the target environment, and these need to be written down.

Dissemination of wisdom is one more reason for having a standard. Rules learned the hard way can be passed on so that inexperienced programmers or those new to the language can quickly get up to speed.

Programmers are a sensitive bunch, and resent having their freedom of expression curtailed

Finally, it is desirable to encourage uniformity across all the source code. Readability apart, measurement plays a crucial role in large development projects. We need code metrics to monitor progress in any meaningful way and to adjust time-scales as work progresses. As measurement is usually based on counting lines of code, it is vital that each programmer is writing lines of about the same complexity.

Read Me

Back in the sixties, when Dijkstra was inventing semaphores and rubbishing GOTOs, FORTRAN-66 was the state of the art in programming languages. Necks, however, were craned towards the new block structured languages which were appearing over the horizon. By allowing programs to be expressed in a more natural way, these promised, among other things, dramatic increases in program clarity.

But just getting rid of GOTOs isn't enough. To be effective, a programming standard has to fight chaos on many fronts, extending the language by use of convention as well as plugging holes in its syntax. Clarity is the overriding concern, and each rule should make a definite contribution. Good inline documentation, straightforward coding and a well defined source file structure are the guiding principles.

Defining a rational layout for source files is a good starting point for any programming standard. This should say what things go in which file, and include a systematic naming convention for files so that components can be tracked down easily. It is usually considered wise, for example, to separate `#defines` from *bona fide* C code, so confining constant and macro definitions to header files may be made a rule.

This is also where thorny problems such as `#include` file management are tackled. Policy should be established early to avoid things like recursive `#include` statements, which often hide away until integration time. The distribution of code around source files can have serious implications (segment hassles under MS-DOS, for example), so it's worth applying expert knowledge to its design.

Once the source file distribution has been sorted out, a skeleton for the inline documentation must be considered. This usually starts with a standard template inserted as a comment at the top of each file, with other templates in the body of the file to document data structures, groups of constants and so on.

Arguably the most important template is for a header which appears before individual functions. Function headers are where most of the inline documentation goes, and it is best to have one for each function. Apart from the obvious things like a description of the function, its calling interface, global data accessed and so on, this is a place to put anything that might be useful to someone trying to understand what the function does or how it works.

Minimal commenting inside the functions themselves should be encouraged. Self-documenting code is preferable, and a function header should never be too far away, so important notes can go there.

NDOWS NOW SUPPORTS WINDOWS 3.0 AND MICROSOFT C6.0 NOW SUPPORTS WINDOWS 3.0 AND MICROSOFT C6.0 NOW SUP

db_VISTA III™

Database Management System

FEATURES	YES	NO
Single and multi-user available	✓	
Relational B-tree indexing	✓	
Network database model	✓	
Multiple database access	✓	
Referential integrity	✓	
Transaction processing	✓	
Automatic recovery	✓	
Record and file locking	✓	
RAM resident		✓
Relational Query and report writer	✓	
Total database redesign/restructuring	✓	
C compilers: most supported	✓	
C ++ compatible	✓	
Operating Systems: VMS, Ultrix,	✓	
UNIX, BSD, SunOs, XENIX, QNX,	✓	
MSDOS, MS Windows and Macintosh	✓	
OS/2 compatible	✓	
LANs: Netbios and Appleshare	✓	
Read and write WKS, WK1 and DBF files*	✓	
Source Code available	✓	
Training courses available	✓	
Run-time Royalties (Absolutely NOT)		✓
*using WKS Library		

The db_VISTA III™ database development system is intended for use by the professional C applications developer. db_VISTA III is written in C and provides a complete set of sophisticated development tools that feature:

- High speed access to large or complex data
- Mainframe functionality
- Portability to any C environment
- Royalty-free run-time distribution
- Source code available in C.

db_VISTA III provides both Relational and Network model technology for programming flexibility. Retrieve a record using the relational keyed access method and all related records can be immediately available using the network database model.

db_QUERY provides SQL-like access to **db_VISTA III**, callable from within an application, **db_REVISE** allows restructuring of live databases.

db_VISTA III is fast. Access times for retrieving data are largely independent of the size of the database.

db_VISTA III is portable. Develop an application in any environment and your database access code can be ported without change to any other supported environment. Currently, **db_VISTA III** supports MSDOS, MS Windows, OS/2, Macintosh, QNX, XENIX, UNIX System V and Berkeley, Ultrix and VMS.

db_VISTA III is a product of Raima Corporation and has been sold to thousands of C programmers in over 50 countries. **db_VISTA III** is fully supported in the UK and Eire by Systemstar Ltd. Comprehensive training is available together with bespoke system development. At Systemstar we offer **C-scape 3™** screen interface library with Look and Feel™ from Oakland Group Inc to provide **db_VISTA III** programmers with a complete application development environment.

For more information about **db_VISTA III, db_QUERY, db_REVISE** and **C-scape 3** call Systemstar in Hertford on **(0992) 500919.**

SYSTEMSTAR
LIMITED

1-3 PARLIAMENT SQUARE HERTFORD SG14 1EX
TELEPHONE: (0992) 500919 FACSIMILE: (0992) 554261

CIRCLE NO. 857

The use of prefixes on identifier names can help track down code from different modules. For variables, the so-called 'Hungarian' system can dramatically reduce bugs attributable to weak type checking. This uses single letters as codes to represent basic types, and each variable carries a prefix (or suffix) made up of a combination of such codes. Thus, a pointer to an unsigned integer might be called `pu_index`, where `p` denotes a pointer and `u` an unsigned integer. This leaves no room for ambiguity when reading or modifying code.

GOTO Hell

With a basic infrastructure in place, attention can be turned to the code itself. The number of spaces to indent and the shape of a `switch` statement are less important than that everyone uses the same format. Most important are rules which enforce plain, straightforward coding. Idiomatic or obscure code can rarely be justified, except where hand optimisation is a genuine necessity.

This is the point where many developers get off. Programmers are a sensitive bunch, and resent having their freedom of expression curtailed. Aesthetics certainly have a part to play in programming, but the actual coding is more of a craft than an art.

Credibility dictates that policy on coding restrictions must be comprehensive and well thought out. Many standards blandly prohibit `break` and `continue` statements, for example, while saying nothing at all about side-effects in expressions or the propagation of error conditions through nested function calls. The humble GOTO statement is the greatest source of ill-considered hysteria, and usually gets star billing in token programming standards. Figure 1 shows a legitimate application of GOTO.

The humble GOTO statement is the greatest source of ill- considered hysteria

The Standards Police

It goes without saying that a programming standard should be put together from experience. Inviting programmers to contribute is the best way to accumulate wisdom, but doing a superficial rehash of a standard pilfered from elsewhere can be a quick way of getting into trouble. In a project-based environment, it is all too easy for a mediocre standard to propagate, made legitimate at each step by an official project cover sheet.

```
#define EXCEPTION_IF(cond, label)  {if (cond) goto label;}
#define EXCEPTION(label)            label
#define EXIT_BLOCK                  exit
#define EXIT                        goto exit

RCODE delete_node(NODE *list, int key)
{
    RCODE rval;

    rval = OKAY;

    EXCEPTION_IF(list == NULL, e_empty);

         :  (search for node in list)

    EXCEPTION_IF(target == list, e_notfound);

         :  (delete node from list)

    EXIT;

    EXCEPTION(e_empty):
        DEBUG("Deletion from empty list");
        rval = E_EMPTYLIST;
    EXIT;

    EXCEPTION(e notfound):
        statistics.miss++;
        rval = E_NOTFOUND;
    EXIT;

    EXIT_BLOCK:
        return(rval);
}
```

Figure 1 - GOTOs have their uses

A key point about a programming standard is that it should be mandatory. It is no use laying down a set of carefully researched rules and then allowing people to ignore them. On the other hand, doggedly enforcing the letter of the law at all costs may not be appropriate. No rule is absolute, and exceptions should be granted where there is justification.

Enforcement of programming standards can be incorporated into whatever quality control mechanisms already exist for a project, and some rules can even be enforced automatically by software tools. The worst kind of tool in this class is a 'pretty printer', which takes code in any old format and adjusts the layout according to pre-defined templates. Although pretty printers just encourage sloppy code, they have limited use for savaging poor quality third-party source.

The CodeCheck tool from Abraxas Software is worthwhile if portability is a requirement. This can check conformance against many different standards (ANSI, K&R, VAX etc etc), as well as run code metrics and check Hungarian prefixes. CodeCheck is completely programmable to cover any other rules you might want to impose.

Your Flexible Friend

The road to Hell, they say, is paved with good intentions. A carefully drafted standard engraved on stone tablets and applied dogmatically can be just as bad as a token 'advisory' standard. From the comfort of your word processor it's tempting to believe you've figured all the angles, but even the simplest rules can develop warts in the field.

You should expect problems to surface once coding is under way, and have a plan for dealing with them expediently. This means having a good feedback mechanism so problems get reported promptly, and being prepared to spend time sorting them out. The danger of half-hearted after-care is that different *ad hoc* solutions will spring up, defeating the purpose of the standard.

Chaos has no place in the software development cycle, and anything that helps to keep it at bay is a good thing. A good programming standard won't guarantee quality, but it will certainly improve the chances of achieving it.

EXE

Mark Hurst works for Logica Communications as a programmer and analyst.

SEE US AT

SOFTWARE TOOLS

Software Tools '91

11-13 June
Wembley Conference
and Exhibition Centre
London

LBMS Systems Engineer.

Multi-user CASE to boost your team's productivity.

It's here! Practical CASE automation that takes team productivity to a new high . . .
LBMS Systems Engineer. Built on the LBMS pedigree, this multi-user windows-based tool offers leading-edge technology to support the rapid application techniques you want to use.

So why multi-user? Systems Engineer networks your team to provide concurrent and immediate access to shared information. The resulting accuracy and consistency are the keys to rapid progress in application development.

How does Windows help? Windows™ 3 is a breakthrough in performance and usability. Systems Engineer fully exploits this natural desktop approach. As a result, multiple techniques can be used side-by-side to achieve the ideal decision-making context for quality designs. Systems Engineer transforms the PC into a powerful developer's workstation by integrating CASE with your choice of Windows™ tools (word processing, e-mail, project management etc.). For added flexibility, an OS/2™ version will be ready when you are.

As an IBM AD/Cycle™ vendor and partner to other leading industry suppliers, we offer an integrated set of CASE and methods solutions that can be matched perfectly to your development environment.

So that you gain the very best from our CASE tools and methods, we back you with the level of training and support that's appropriate to you. Our aim is to transfer the skills and experience of the best people in the business to your organisation.

Get LBMS integrated CASE solutions working with your team. Phone Elizabeth Baxter now on 071 636 4213 for a demonstration, more information or details of our CASE technology seminars.

Evelyn House 62 Oxford Street London W1N 9LF
Tel: 071-636 4213 Fax: 071-636 2708

LBMS

CIRCLE NO. 858

Trademark ownership. OS/2™, AD/Cycle: International Business Machines Corporation. Windows™ : Microsoft Corporation.

An old friend in new clothes

Microsoft's new Visual Basic is claimed to be the 'easiest way to create and integrate Windows applications'. Will Watts was granted a pre-release copy.

Microsoft has been a little quiet of late. While other companies have been launching OOP and Windows utilities galore, the great MS has slumbered, occasionally stirring to make promises about tools to be released at some unspecified future date. The release of Visual Basic, a Windows-based general purpose development tool, marks the return of MS to the attack. Does VB meet its manufacturer's extravagant claims of being the Windows tool for all?

How it works

VB offers a novel kind of programming environment, which merits a careful description. The VB environment is run up by double-clicking on an icon which materialises in the Program Manager Window during installation. Figure 1 shows the environment that appears, loaded with Microsoft's sample calculator application. The main window, at the top of the picture, contains the main menu bar (File, Edit, Code etc) and the Properties Bar. The multi-buttoned window on the left hand side, the one that looks a bit like a grey bar of chocolate, is the Tool Box. The window with the calculator and a grid of dots is the current form - you edit this to become your application's interface. There is also a project window, which lists the files in an application, and a palette window for selecting the colours of control components. All these windows (and a deal more, which we haven't opened yet) have the run of the whole screen; there is no 'master' window which surrounds the lot. Microsoft has abandoned the MDI system of having child windows shut inside a single parent, as implemented in, say, Turbo Pascal for Windows

The first stage in writing a program is to set up the user interface, using the default form and the Tool Box. The buttons on the Tool Box represent standard Windows controls: Picture box, Label, Text box, Frame, Command button, Check box, Option button, Combo box, List box, Scroll bars h & v, Timer, Drive list box, Directory list box and File list box. The Timer requires additional explanation: It is a non-displayed object which gives the VB programmer access to Windows' `WM_TIMER` messages - equivalent to hanging a piece of code off the clock interrupt in DOS terms. However, this facility is much more important in an event-driven environment, where you cannot achieve, for example, animation by writing a timer loop.

To add a control to a form, you select the relevant toolbox button and click the mouse at the chosen spot. Once deposited, the item can be moved around and resized - the latter is achieved by adjusting its 'sizing handles' (a set of draggable black blobs which appear at the corners of the control when it is selected). To help you line up controls, VB incorporates a 'snap' feature - ie the object automatically remains aligned to the background grid. Snap can be turned off if not desired, and the density of the grid is adjustable. The dimensions of the current object are actively displayed on the right hand side of the Properties Bar. Which brings us to...

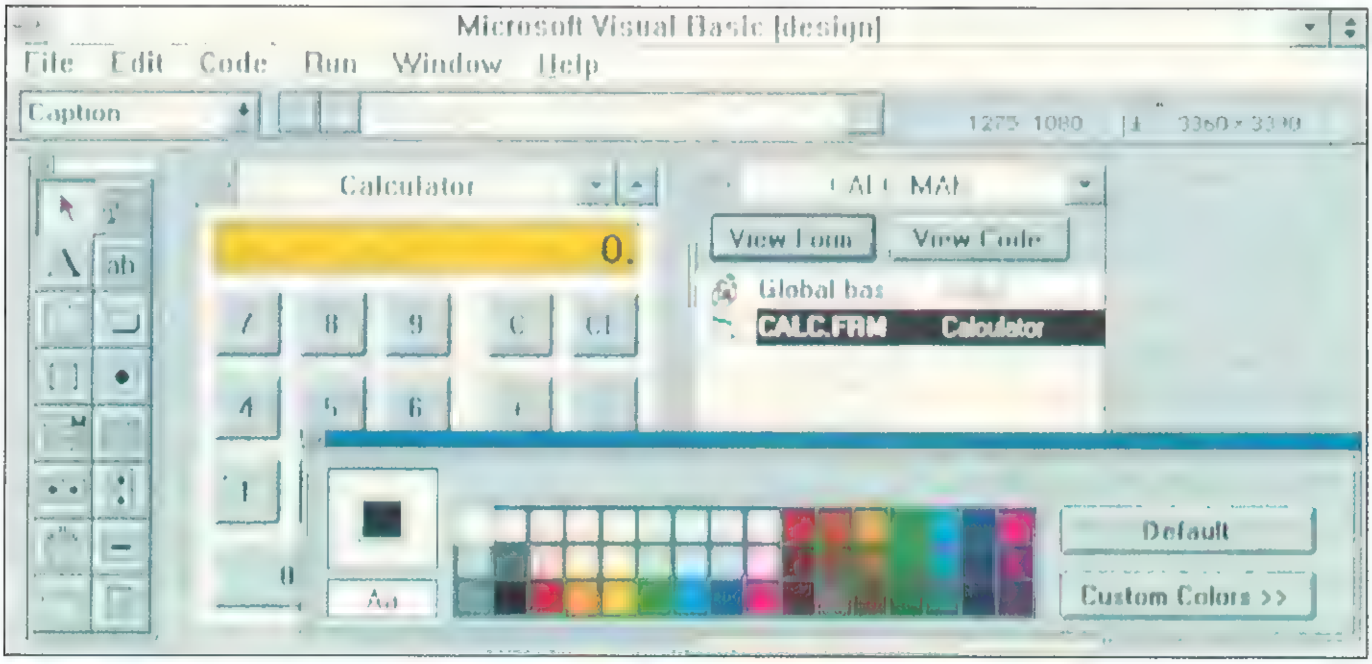


Figure 1 - The Visual Basic Environment

PRO-DIR V3.0

The Power Directory Utility for the Professional Programmer

NO MENUS !!!
45 page manual and on-line help included

PRO-DIR brings you the power of a command line directory utility with unsurpassed search and selection features. It combines some of the best features seen on DOS, UNIX, VMS etc. at an affordable price of just £49 (including VAT and P&P).

PRO-DIR makes the task of finding and managing files effortless, even on large directory structures. You can now find those long lost files on your network drive with commands like

```
      d  G:\USER\BOB\...\TAX*R*T*N.*     after=01-01-87     before=-365
```

Look how easy it is to recover disk space when your hard disk is full and you need to free a few megabytes quickly - the command

```
      d  C:\,D:\,E:\     +R     min=50k     since=-7
```

finds all files larger than 50kb that you created over the last seven days *anywhere* on drives C, D or E. (Also see below how the +B switch can help).

TAKE A LOOK AT THESE FEATURES

d *A*B*02.E?E	Extended wildcards in all file and directory specs
d *.EXE,*.BAT,*.COM	Multiple unlimited file *select* specs
d *.* *.EXE,*.BAT	Multiple unlimited file *exclude* specs
d C:\**S*R*C**.C	Extended wildcard directory specification
d C:\*.C,D:\USER\*.C	Searching on multiple drives (network drives are supported) and directories
d C:\USER\*.C +R	Recursively find all matching files below specified directory
d ***.C	Search for files that are a specified number of levels deep
d \A*\...*S*R*C*\...*.C	Search for files with an unknown number of intervening sub-directories
d +S:f	Sort by filename (or file extension, size, date, attribute)
d max=2M min=60k	Select by file size range
d on=-7	Select by relative (or absolute) date. Relative dates are from current date
d after=-365 before=-31	Select by relative date range
d after=10-12-90	Select by absolute date range
d a=h n=d	Select/exclude by file attribute (eg d=directory file, h=hidden file)
d +p	Comprehensive display paging options
d +B	Generate output in batch file format for further processing
d +T	Display the total disk usage of selected files
d +h	Display hidden files
d +m	Display the seconds field of the selected files
d +ft-asd	Enable or disable display fields. (eg f=filename field, s=file size field etc.)
d +?	Display the on-line help screens
set PDOPTIONS=+pt-h	Complete customization of program default settings via the set command

You can combine all these features together. You may specify as complicated a command as you wish, provided it fits on the DOS command line! And that's not all - we haven't even listed all the switches and options that are available. Call or fax us for more information. Alternatively, send us the coupon below, or call us with your order 9AM - 5.30PM.

Please send me ____ copies of PRO-DIR at £49 per copy including VAT and P&P.
Name: ....
(Company): ....
Address: ....
....
Access/Visa number: .... Expiry date ....
Please specify disk format: ____ 5.25" (360kb) OR ____ 3.5" (720kb)
If paying by cheque, please make it payable to Korala Software. Thank you

KORALA
SOFTWARE
Korala Associates Ltd
FREEPOST
Edinburgh EH3 0EP
Phone. 031 556 9208
Fax 031 556 9215

Suitable for IBM PC, XT, AT or 100% compatibles running DOS 3.0 or higher. Please allow 28 days for delivery. International orders: No VAT, postage charged at cost

CIRCLE NO. 859

PROTECT YOUR PROGRAM FROM PREDATORS

- Established for over eight years in software security products
- Internationally established product
- Competitively priced keys (from £18.00)
- Supports IBM PC/XT/AT and PS/2
- Supports Apple Macintosh range
- Advanced manufacturing techniques
- Transparent and user friendly
- No drivers necessary (with exception of programmable key)
- Large range of security keys.

For further information and 60 day evaluation kit, contact:

COMSEC (U.K.) LTD.
Cranfield Airport,
Cranfield,
Bedfordshire,
MK43 0AL
Tel: (0234) 751203
FAX: (0234) 751082

COMSEC
(U.K.) LIMITED
COMPUTER & COMMUNICATIONS SECURITY SYSTEMS

CIRCLE NO. 860

Development speed = Oregon C++

Oregon C++ is, quite simply, the fastest C++ Development System software you can buy. Consisting of a true optimising compiler (not just a C translator), a source level debugger and libraries, it sets speed records in every area. Here's how.

The optimising compiler means optimum speed

Because the optimising compiler directly generates compact object code it eliminates the translating step. The result? Faster compilation, direct debugging and faster program development. And you end up with smaller and faster applications. What's more the compiler is switch selectable for C++, ANSI C, or K&R C.

You don't waste time worrying about compatibility

Oregon C++ conforms to ANSI standards for the C++ language and supports inter-language calls to and from C, Pascal, Modula-2 and Fortran. So you get fast, easy, access to existing code modules, without wasting time rewriting or re-debugging. And Oregon C++ is totally compatible with all existing C libraries.

Fast debugging is done in the same language in which the application is written

The Oregon Debugger - ODB - debugs C++, ANSI C and K&R C in the original application language, which means you get more reliable code in less time. And because the ODB understands multiple inheritance, it can quickly display the class hierarchy.

Oregon C++ is fast and easy to use

With a choice of command line or mouse-driven window user interfaces, Oregon C++ is easy to use. In window mode you even get separate windows for the application and debugger I/O, and every window is fully scrolling so you can find the information you want in no time at all.

Fast information...

For an instant response to your request for information, call us today. Or clip the coupon for your free technical data.

Instrumatic®
The Pan-European Technology Group
Instrumatic UK Ltd.
First Ave., Globe Park, Marlow, Bucks. SL7 1YA
Tel 0628 476741 Fax: 0628 474440
Tlx 847042 IMATIC G

For more information fill in the coupon and post today!

Name
Position
Company

Postcode Telephone

.EXE 6/91

CIRCLE NO. 861

Properties

With the form laid out, the next move is to edit the properties of the controls. As will become clear, the controls from the toolbox are more or less OOP objects. Their properties are more or less equivalent to what OOP programmers would call instance data or attributes.

Each control's property list is pre-determined, and is non-extensible. Some properties are obvious. Labels and buttons have `Caption` properties which are the strings that they display. The form itself has properties: `Caption` (again), `CtrlName`, `Cursor`, `Default`, `DragIcon`, `Enabled`... The `CtrlName` property, common to all controls, is of particular importance, as this is used in the Basic code to identify the object. All properties are assigned default values on creation, for example, the `CtrlName` and `Caption` properties are set to `Label1`, `Label2`, `Label3` etc.

An object's properties are set by selecting the control in question (clicking on it), then activating the properties list box (LHS of Properties bar). Choose the property that you wish to adjust from the list. The middle box, the 'Settings' box, now contains the current value of that property, and may be set by typing in the new value.

At last, some code

So far, then, very much like any interface design tool. We now have a form, decorated with suitable controls, with all the text labels etc set up. It is possible to run the application in this state: the buttons will 'push', the scroll bars slide, and the input boxes will accept text. But nothing useful happens. Time to make the program *do* something.

You add functionality to a VB application by writing *event procedures.* In the same way that each object type owns a predefined list of properties, it also has a set list of event procedures. The list corresponds to a set of events (or messages, if you wish to take a lower-level perspective) which the programmer can choose to act upon. For example, the list for push buttons includes `Click`, `DragDrop`, `DragOver`, `GotFocus` and `KeyPress`.

To add program code, you must open yet another window - the Code window (see Figure 2). This is essentially a text editor, with various bells and whistles attached. Suppose that you wish to specify the action which should occur when the operator presses a button christened `TestCmd`. The left hand field in the bar at the top of the Code window, 'Object:', allows you to select `TestCmd` from a list of all the objects in your form. The right hand field, 'Proc:', has a similar list of all the methods. When the object field has been set to `TestCmd` and the proc field set to `Click`, the editor will contain

```
Sub TestCmd_Click ()

End Sub
```

- a template for the procedure that you are about to write. (Other procedures have more complex templates, for example, `KeyDown` has `Sub Command1_KeyDown (KeyCode As Integer, Shift As Integer)`). All navigation around VB's code is like this. To inspect the code for another object or procedure, you must repeat the palaver of selecting from the list controls in the bar at the top. Initially this feels very odd. There is no concept of the program to a single body of text. There is no paging through acres of code, looking for a particular little snippet.

The editor's other great trick is to perform parsing-as-you-go tricks. Type in a line `? i%` then cursor down and blink, you find that it has substituted your common slang with the Queen's Basic: `Print i%` (I know GWBASIC has been doing a similar sort of thing for 4000 years, but VB is heaps slicker.) It also picks up syntax errors, and won't let you leave the line until you have either fixed the mistake or got mad and commented the whole damn thing out.

The language

The Visual Basic language is not a fully-fledged OOPL. There is no mechanism for creating your own classes, or inheriting. The frames and Toolbox controls are pseudo-objects, but you are restricted in what you can do; in effect you can only alter instance data and override methods. The properties (= instance data) can be accessed using the standard dot syntax, for example, `TestCmd.Caption = "Test me"` VB uses a strange naming convention for its method-like procedures: *<object id>_<proc id>*. The object and method identifiers are glued together into one name (the maximum identifier length has been set at an unusually long 40 characters to accommodate this). One might guess that the normal dot syntax for identifying methods has been reserved pending the arrival of genuine objects in Microsoft Basic.

The limitation of being restricted to using objects that you have drawn on the screen at design time is heavy. For example, with the system as described so far, you cannot create a menu containing a variable number of items. Microsoft has provided an interesting, but rather smelly, mechanism to get round this. What you do is define arrays of objects. At design time, you define one object and set its `Index` property to the number which you wish to be the base of your array (ie '0' if you are a right-thinking C programmer). You now have a single-element array. At run-time, you can add and delete new elements to this array, using the keywords `Load` and `Unload`. So `Load AppName(1)` creates index position 1 of the object array `AppName` - remember that Basic uses round brackets for array elements. What is happening is that you are being assigned memory for another set of instance data. The new control object automatically gets the methods assigned to the base element - it's just like instantiating a new object in a real OOPL. These methods are passed the index of the current object, so they can work out who they are working for. It's down to you to track which indexes you have filled, and if you should be rash enough to try to `Unload` the element that you put in at design time... Well, I told you it was smelly.

VB provides access to all the standards of Windows programming: fonts, bitmaps, icons, the clipboard, DDE. As suggested above, it also supports the `PRINT` statement (prints to the current form), although more Window-ish forms of output, such as message boxes, are encouraged.

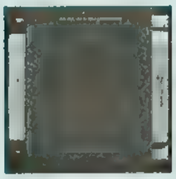
Other Basic language issues: if you know QuickBASIC, then the syntax and facilities of VB are very similar, so you can skip this bit. If you last touched Basic when it had line numbers, here is a 10 second tour. Line numbers are dead, but tolerated if you insist. Control structures are greatly improved, in particular with the introduction of `If..End If` blocks (avoiding dangling `ELSE`s) and a `Select Case` statement replacing the vile `ON...GOTO`. Functions and procedures are now supported with full parameter passing and a weird syntax which is as orthogonal as a cold chip. Proper scoping has been introduced. You are now encouraged to declare variables before you use them. Variable typing using, eg a `%` suffix to signify integers and a `$` for strings, is still there. Strings remain great fun, provided you are not doing mixed-language programming. Oh yes. And BASIC has been renamed Basic, and we are now allowed lower case letters in our program code.

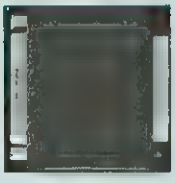
Program organisation

I have got this far, but I have still yet to save my work! VB works with three types of file: forms, which hold the graphical objects and (binary format) 'method' code associ-

SOFTWARE PROTECTION!

Can You Spot The Difference?





- ☑ High-level security keys
- ☑ Assembler-based, customer specific, encrypted interrogation routines
- ☑ Over 70 languages supported
- ☑ Compatibility - due to 8 years experience & 600,000+ keys sold
- ☑ Reliable on-going support
- ☑ Physically unique keys for each customer
- ☑ MS-DOS, OS/2, UNIX, XENIX, WINDOWS 2 & 3, "MACINTOSH"
- ☑ Transparent operation with most peripherals on market
- ☑ Free language updates
- ☑ Parallel, Serial, Mac ports

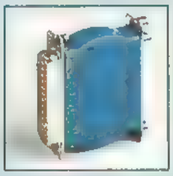
OVER 3,100 CUSTOMERS WORLDWIDE HAVE!

Electronic Key

The ideal device for identically produced software packages. Uniquely wired with customer code and a software code. Uses Assembler based program, decryption interface and random values.

1 Word Memory Key

Custom hardware wiring allows the developer total control over information stored in the key. 2 bytes of memory allows several packages to be protected with just the one key.

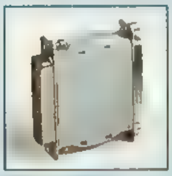


31 Word Memory Key

For multiple software protection schemes. 31 words of 16 bits of non-volatile dynamically programmable memory. Its capacity to store information provides virtually limitless power. Flexible protection scheme can be modified on-site during operation of software package.

Micro Processor Key

Provides the ultimate in software security. Not tied to any language or O/S. 8 bit microprocessor powers from RS-232 level. Requires no power supply. For PC terminals, minis, & others using RS 232 C comms. Used on workstations. This key is effectively a computer.



Macintosh Key

Extremely powerful & customised protection for the Mac. 31 words available for random storage. Providing unequalled protection the Macintosh Memory Key connects to the SCSI 25 pin port and operates transparently.

MICROPHAR... The European Standard!

For a no-nonsense informative discussion on how our dongles can better protect your profits, please call us on:

091-378 9191

CLEARSOFT

CLEARSOFT, Littleburn Ind. Estate,
Langley Moor, Durham DH7 8HG.
Telephone: (091) 378 9191 Fax: (091) 3789393

Please send me full details on your protection systems.

Name: ....

Position: ....

Address: ....

....

Telephone: ....

Fax: ....

Signature: ....

Clearsoft, Littleburn Ind Est,
Langley Moor, Durham, DH7 8HG
Tel: 091 3789191 Fax: 091 3789393

CIRCLE NO. 862

ated with them; modules, which contain any general-purpose code procedures that can be used by the whole project; and Project Make files, which hold a list of all the forms and modules in the project. From this you will gather that applications may contain multiple forms.

VB is auto-compiled as you enter it, so it can be run almost without a pause (although there *is* a pause, while the environment tidies up the various windows). It has full debugging facilities - break points, single step and examination of run-time variables. The latter is achieved via the use of an immediate window, into which you can type commands such as `? i%`. As was pointed out to me by a Microsoft rep, this is not as good as having a proper watch facility which continuously monitors selected variables, but it's not bad.

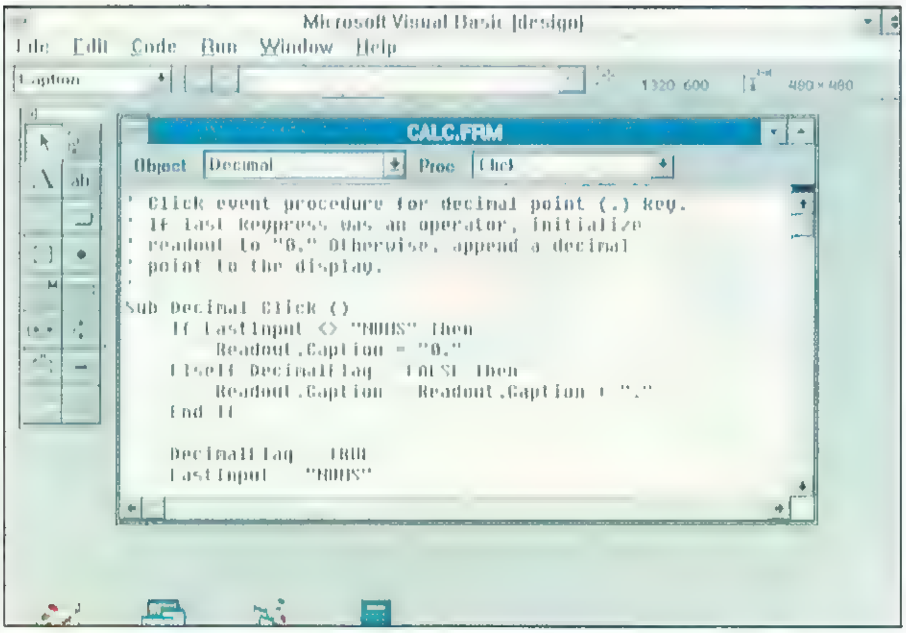
VB can produce stand-alone .EXE files for distribution. These do require the presence of a run-time module, but this may be distributed without license payments, whoopee! There is no facility for producing DLLs.

Other matters

I'm out of space, so I must race through a few other items of importance. The package is bundled with a VB tutor program, which is very pretty, but probably too slow for experienced programmers. Also bundled is the source for an icon editor application - you can see it running in Figure 3 - and a large library of icons. Online help is context-sensitive and excellent. I did not have a complete copy of the printed documentation for this preview; but what I had suggested that MS has abandoned the policy of skeleton offline documentation (as used in C 6.0) in favour of proper manuals. The stuff that I had was well-written.

Microsoft is offering various add-ons for VB. These consist of the Help Compiler (as found in the SDK), the programming reference book and online resource (allows VB programmers to access the Windows API directly, which must be something of a pain in a language that doesn't support pointers), libraries for the SQL server (allows VB applications to front-end the Microsoft SQL server) and a Control Developers kit. This last allows C programmers to add their own controls to VB's toolbox, by writing special DLLs. Thus VB has a kind of extensibility. I have some of the gen on doing this; I must warn you that it looks like hard work. MS says that third parties are also toiling over VB add-ons and an OS/2 version of VB is threatened.

One further point: programmers of COBOL and FORTRAN will be able to write DLLs and use VB to generate the Windows interface of their applications, so nobody will be left out of the fun.


Microsoft Visual Basic [design]
File Edit Code Run Window Help
Caption
1320 600
490 × 480
CALC.FRM
Object Decimal
Proc Click

```
' Click event procedure for decimal point (.) key.
' If last keypress was an operator, initialize
' readout to "0." Otherwise, append a decimal
' point to the display.
'
Sub Decimal_Click ()
    If LastInput <> "NUMS" Then
        Readout.Caption = "0."
    ElseIf DecimalFlag = FALSE Then
        Readout.Caption = Readout.Caption + "."
    End If

    DecimalFlag = TRUE
    LastInput = "NUMS"
```

Figure 2 - The code editor

ADVANCED BUSINESS GRAPHICS

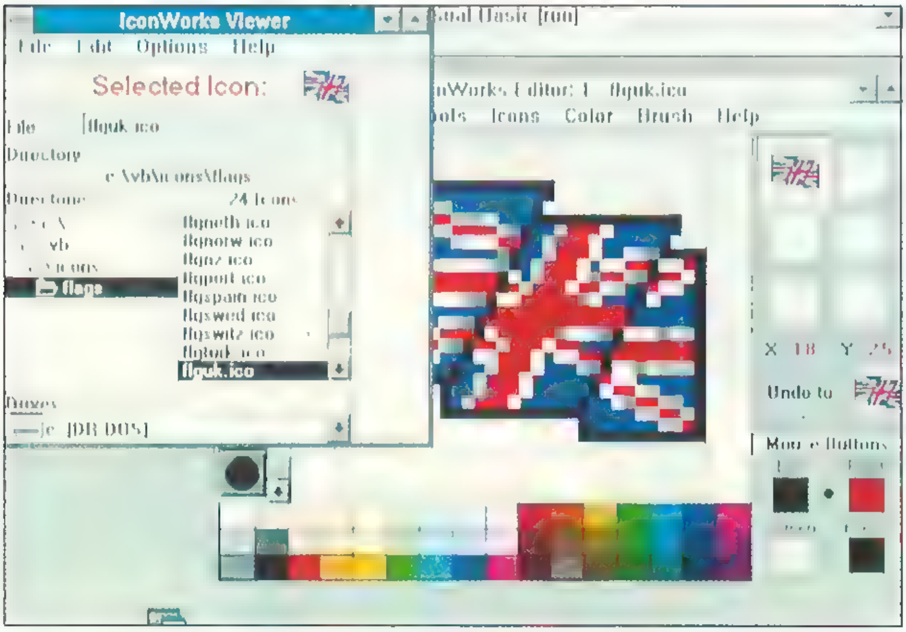


Figure 3 - The IconWorks demo application

ADVANCED BUSINESS GRAPHICS

Conclusion

GUI applications are easier to use than describe. If I have given the impression that VB is hard work, then this is the clumsiness of my prose, because it's a cinch.

Cut open any 4GL or application generator, and eventually you hit the point where a trade-off has been made between ease of use and flexibility of the system. This point is well-buried in VB. The limitations of this BASIC running under Windows are the same as ordinary BASICs: not so hot on linked lists and low-level stuff, but pretty good for common or garden applications.

Microsoft envisages a wide range of uses for VB: from prototyping complex applications, the creating of 'medium sized' programs, down to hacking about as a kind of Windows batch language. Given the excellence of the product, and the aggressive pricing level, I reckon that it is right. EXE

Visual Basic is priced at £139, or as a £79 upgrade to registered users of other MS BASICs. The add-ons mentioned above cost £39 each, except for the SQL Server library, which is £TBA. The software should be shipping by mid-June. Microsoft is on 0734 500741.

RECESSION SPECIAL! TAKE 20% OFF ALL PRICES!

PRE-SALE CONSULTING

POST SALE SUPPORT

NEW AND NOTEWORTHY:

Brief 3.1. ...New Version!..	£ 194
Greenleaf Comm++....	£ 164

True C++ Class Library, Device independence through inheritance. Baud rates to 115,200

LANtastic 2mbps Starter Kit ...	£ 362

Economical alternative to Novell. Link as few as 2 PCs and grow up to 100s.

MS-DOS v5.0 upgrade!...Finally here!	£ 80
MS Macro Assembler v6.0 .New Version!.	£ 107
Novell Netware v2.2	£ 605

Combines Netware v2.x SFT, Advanced, ELS I & ELS II into one product

Novell Netware v3.11...	£ 2355

Complete re-write Take full advantage of the 386 chip.

PC Tools v7.0......New Version!	£ 112
SQLWindows for Btrieve V2.0	£ 1486
Turbo Pascal for Windows	£ 174

Create, edit, compile and run programs, all from within Windows!

Win++	£ 176

Nearly 100 classes provide a simplified, flexible & powerful C++ interface. Make your Windows development a breeze!

ARTIFICIAL INTELLIGENCE

Arity Combination Pack v5.1	£ 594
PDC Prolog (Turbo Prolog) v3.21	£ 170
PROLOG-2 Personal v2.15	£ 81

ASSEMBLERS & LINKERS

.RTLink/Plus v4.1	£ 270
Blinker v1.11	£ 130
Microsoft Macro Assembler v6.0	£ 107
Plink86+ v2.3	£ 216
Spontaneous Assembly v2.0	£ 150
Turbo Debugger & Tools v2.0	£ 99

BACKUP

Back-It v4.0	£ 65
Central Point Backup v6.0	£ 49
Copy II PC v6.0	£ 28
Copy II PC Deluxe Option Board v5.4	£ 94
Norton Backup v1.2	£ 95

BASIC

Microsoft BASIC Pro Dvlpmnt System v7.1	£ 258
Microsoft QuickBASIC Compiler v4.5	£ 54
PowerBASIC v2.1	£ 61
Stay-Res Plus v4.0	£ 74

BUSINESS

ABC Flowcharter v1.1	£ 151
Form Tool Gold v3.0	£ 48
Grammatik Windows.	£ 55
Instant OrgChart v1.0.	£ 101
Lotus 1-2-3/G v3.1	£ 515
Micrografx Charisma v2.0	£ 274
Microsoft Excel-Windows v3 0	£ 252
ORG PLUS Advanced v5.0	£ 53
PerFORM Designer & Filler Windows	£ 310
Quattro Pro v2.0	£ 261
SideKick Plus v1.0	£ 107
Super Project Expert v2.0.	£ 354
TimeLine v4.0	£ 371
Xerox FormBase Windows v1.1	£ 252

C

Aztec C Cross Development Systems	£ 317
Borland C++ v2.0	£ 325
C ASYNCH MANAGER v3 0	£ 104
C-terp 386 v3 5	£ 217
CLEAR+ for C v1 0	£ 103
Clipper v5.01	£ 539
Control Pak/Windows	£ 222
DESQview API C Library v1.3	£ 97
DESQview API Toolkit.	£ 232
Essential Communications v3.0	£ 165
.Essential Graphics v4.0	£ 324
graphicsMENU v3 0	£ 115
Greenleaf ComLib v3.1 Level 2.	£ 270
Greenleaf ViewComm v3.0	£ 439
Guido v1.5	£ 130
Intel 386/486 C Code Builder Kit v1 0	£ 371
Lattice C Development System v6.0	£ 137
Microsoft C Compiler v6.0	£ 258
Microsoft QuickC v2 5	£ 54
Microsoft QuickC with QuickAsmblr v2.5	£ 105
Paradox Engine (C version) v1.0	£ 154
PC-Lint v4.0	£ 65
Pro-C with Workbench v2.0	£ 470
Turbo C TOOLS v6.02	£ 83
Vitamin C for Microsoft C v4.0	£ 124
WATCOM C /386 Professional v8.0	£ 760
Zortech C++ Developer's Edition v2.1	£ 245

CADD

3D Drafting	£ 169
AutoCAD Release 11	£ 2390
Design CAD 3D v3.1	£ 164
DRAFIX CAD for Windows v1 1	£ 346
Generic CADD v5.0	£ 191

CASE

Case: PM for C++ v2.1	£ 1510
Case: Windows v3.0	£ 754
Compeditor II v3.09	£ 191
EasyCASE Plus	£ 215
Teamwork/PCSA v3.1	£ 499

COBOL

Microsoft COBOL Compiler v4.0	£ 467
RM/COBOL-85 Dvlpment System v5.0	£ 636

COMMUNICATIONS

Carbon Copy Plus + Host v5.2	£ 92
Crosstalk for Windows v1.1	£ 101
LapLink III v3.0	£ 70
Mirror III v2 0	£ 69
Procomm Plus v2.0	£ 56
Remote2 (Host & Caller) v2.0	£ 90
SmarTerm 240 v3.1	£ 158
Takeover	£ 135

DATABASE & FILE MANAGEMENT

ALPHA/Four v1.1	£ 249
AskSam v5.0	£ 184
Clarion Professional Developer v2.1	£ 383
Clipper v5.0.	£ 423
dBASE IV v1.1	£ 396
dbFast/PLUS v1 4	£ 142
dbFast/Windows v1.5	£ 203
Documentor v1.5	£ 130
FairCom Toolbox Special Edition	£ 378
Gupta SQLWindows v2.0	£ 766
Informix 4GL RDS/ID for DOS	£ 603
Microsoft SQL Server (1-5 users) v1.1	£ 549
Oracle Developers Kit v5.1C	£ 563
Paradox v3.5	£ 412
Paradox SQL Link v1.0	£ 283
Q&A v3.0	£ 188
R&R Clipper and FoxBASE+ Module	£ 29
R&R Report Writer for XDB	
SQL Media v1.0	£ 164
R:BASE 3.1 v3.1	£ 396
Superbase 4 Windows v1.2	£ 353
UI Programmer v2.1	£ 262
XQL Relational Data Manager v2.11	£ 446

DEBUGGERS

Multiscope Debugger for Windows v1.0	£ 230
Periscope II with Breakout Switch v4 0	£ 81
Periscope IV-25MHz v4.0	£ 1520

DESKTOP PUBLISHING

Adobe Type Manager for Windows	£ 48
Astral Picture Publisher v2.1	£ 274
Facelift for Windows	£ 51
Freedom of the Press	£ 203
Go Script Plus v3.0	£ 153
HiJaak v2.0	£ 110
OmniPage /386 for Windows v2.1	£ 353
PageMaker v4.0	£ 415
Publish It! v2.0.	£ 92
Xerox Ventura Publisher Gold v3 0	£ 445

DISK/DOS UTILITY

Disk Manager v4.03	£ 32
Lotus Magellan v2 0	£ 84
Mace Utilities 1990	£ 78
Norton Utilities v5.0	£ 94
OPTune	£ 38
PC Tools Deluxe v6.0	£ 77
Q-DOS II v2.0	£ 31
SpinRite II v1.1	£ 53
XTree Pro Gold v2.0	£ 77

EDITORS

Brief v3.1	£ 194
Brief with dBrief v3.1	£ 275
Sage Professional Editor v1.1.	£ 162
VEDIT PLUS v3 4	£ 110

FORTRAN

F77L-EM/32 v3.0	£ 566
Fortran Toolkit 1 v5.2	£ 60
Fortran Toolkit 2 v5.2	£ 99
GRAFMATIC & PLOTMATIC	£ 132
Microsoft FORTRAN Compiler v5 0	£ 235
RM/FORTRAN Dvlpmnt System v2.44	£ 302
WATCOM Fortran 77 /386 v8.0.	£ 643

GRAPHICS

Adobe Illustrator for Windows v1.1	£ 229
Autodesk Animator v1.0	£ 244
Corel Draw v2 0.	£ 315
Cricket Graph for Windows	£ 101
Designer for Windows v3.01	£ 384
GRASP with ARTOOLS v3.5	£ 94
GSS Graphic Dvlpmnt T/kit for DOS v2.16.	£ 467
GX Graphics v1.0.	£ 137
HALO v3.1	£ 192
HALO Professional v3.0	£ 289
Harvard Graphics v2.3	£ 254
MetaWINDOW v3.7b	£ 188
PCX EFFECTS v1.0	£ 69
PCX Programmer's Toolkit v4.0	£ 170
Pizazz Plus v1.3.	£ 76
StatGraphics v5.0	£ 504

HARDWARE

80387-33 Math Coprocessor	£ 485
ASCII Internal 2400 Modem	£ 59
ASCII Internal 9600bps V.32 with MNP5	£ 361
Expanz Disk Expander Card	£ 108
Practical 2400 Pocket Modem	£ 91
ProScan PS1000 G-scale Full-Page Scanner	£ 412
QuickTel 2400 Internal Modem	£ 51
QuickTel 4800 V 22bis\V.42 Internal Modem.	£ 97
QuickTel 9600 Internal SendReceive FaxModem	£ 146
QuickTel 9600 SendFax Internal Modem	£ 75
QuickTel 9600 V.32bis/V.42 Internal Modem	£ 376
SatisFAXtion Classic	£ 317
ScanMan Plus PC	£ 121
Worldport 2400 Pocket Modem	£ 203

INTEGRATED

Framework III v1.1	£ 358
Lotus Works	£ 77
Microsoft PC Works v2.0	£ 80

LAN & LAN READY

AboveDISC+ (unlimited users) v4.0	£ 164
AboveLAN (unlimited users) v1 0B	£ 203
Disk Manager-N (Novell) v3.04	£ 83
ELS Netware Level I v2.12	£ 446
ELS Netware Level II v2.15	£ 940
LANtastic 10 MBPS Ethernet Adapter v3.0.	£ 176
LANtastic 10 MBPS Ethernet Starter Kit v3.0.	£ 380
Networker Plus v2.2.	£ 119
Saber Menu for MS Windows 3.0	£ 211
Xtree Net ELS Level 1 & 2 v2.0	£ 128

MACINTOSH

Adobe Photoshop.	£ 429
Adobe Type Manager v2 0	£ 49
Carbon Copy Mac Single Pack	£ 94
ColorStudio	£ 504
FastBack v2.5	£ 98
Freedom of the Press v3.0	£ 203
HyperDesk v4.2.	£ 40
LapLink Mac Connectivity Pack v3.0	£ 74
MacSleuth	£ 63
Mathematica for Mac II v1.2	£ 575
OmniPage v2 1	£ 378
Pagemaker v4.0	£ 424
Persuasion v2.0	£ 263
QuarkXPress v3.0	£ 549
Smalltalk/V Mac v2.0	£ 92
StatView II	£ 225
SUM II (Symantec Utilities for Mac) v2.0	£ 82
Think C v4 0	£ 134
Think Pascal v3.0	£ 134

MEMORY MANAGEMENT

386 Max v5.1	£ 61
AboveDISC+ v4.0	£ 50
Headroom v2.02	£ 61
QEMM Expanded Memory Mngr/386 v5.1	£ 50
QRAM v1.0	£ 40
Turbo EMS v6.0	£ 43
Vcache v5.06	£ 40

MODULA-2

EmsStorage with Full Source v1.0	£ 42
Logitech Modula-2 Dvlpmnt System v3.0	£ 134
Repertoire v1.6	£ 69
TopSpeed Modula-2 DOS Compiler Sm v1.17	£ 128

OBJECT ORIENTED PROGRAMMING

Actor for Windows w/Language Extns v3.0	£ 598
Borland C++ v2.0	£ 270
Object Professional v1.1.	£ 113
Object Vision v1.0	£ 75
Smalltalk/V 286 v1.1	£ 92
Zinc Interface Library v1.0	£ 136
Zortech C++ Compiler v2 1	£ 110
Zortech C++ Developers Edition/386 v 2.1.	£539

OPERATING ENVIRONMENTS

386/ix Operating System Release 3 2	£ 243
AIX Advanced Interactive Executive v1.1	£ 467
DESQview 386 v2.3	£ 107
DR DOS v5.0	£ 112
DR Multiuser DOS	£ 479
GeoWorks Ensemble v1.0	£ 90
Microsoft DOS v4.01	£ 59
Microsoft Windows v3.0.	£ 80
MS-DOS v5.0 Upgrade.	£ 80
OS/2 Extended Operating System v1.3.	£ 504
OS/2 Standard Operating System v1.3	£ 213
VM/386 Multitasking v1.22	£ 113

OS/2 PRODUCTS

AsymetrixToolBook v1.0	£ 232
Display Write 5/2	£ 198
MS OS/2 Presentation Mgr.T/kit v1.2	£ 261
OS/2 Programmers' Toolkit v1.1	£ 461
Paradox v2 0	£ 389
PDC Prolog v3.2	£ 407
R:BASE v2.1	£ 494
SideKick for Presentation Manager v1.0	£ 107
Smalltalk/V-PM v1.2	£ 227
TopSpeed Modula-2 OS/2 Compiler v2.0	£ 314
Wide Angle for OS/2	£ 54

OTHER

Dan Bricklin's Demo II (Prototyper) v3.0	£ 137
Guide 3 for PC (HyperText) v3.0.	£ 302
HyperPAD (HyperCard) v2.0	£ 70
Mace Vaccine (Virus Protection)	£ 50
Microsoft Windows Entertainment Pack	£ 23
MKS AWK (Language) v3.1	£ 64
MKS Toolkit (UNIX like tools) v3.1	£ 157
PolyAWK (Pattern matching) v2.3	£ 110
PolyLibrarian (Source documentation) v2.3	£ 137
PolyMake v3.3	£ 91
PVCS Professional (Version Control) v3.3	£ 270
RPG II Development System v3.1	£ 938
System Sleuth (Diagnostics) v2.1	£ 63

PASCAL

Asynch Plus v5.01	£ 104
B-Tree Filer v5.04	£ 75
Graf/Drive Plus Professional v2.0	£ 164
graphicsMENU v3.0	£ 115
Microsoft Pascal Compiler v4.0	£ 158
Microsoft QuickPascal v1.0	£ 54
Object Professional v1.1	£ 113
Pascal ASYNCH MANAGER v1.1	£ 97
Pascal TOOLS and TOOLS 2 v2.5	£ 97
POWER TOOLS PLUS v5.1	£ 83
SkipJack v1.01	£ 183
Topaz v3.0	£ 56
Turbo Analyst v6.0	£ 78
Turbo Pascal v6.0	£ 81
Turbo Pascal Professional v6.0	£ 158
Turbo Plus v5.5	£ 119
Turbo Professional v5.09	£ 78
View 232 v1.0	£ 104

SCIENCE & ENGINEERING

Advanced Math Application Pack	£ 53
ChiWriter v4.0	£ 223
Derive v2.0	£ 155
MathCad v2.5	£ 256
Mathematica 386 v1.2	£ 503
Mathematica for Windows v1.2.	£ 719
Mathpak 87	£ 66
SSP/PC Scientific Subroutine Library v1.0.	£ 207

SCREEN DESIGN & MANAGEMENT

Fansi-Console v3.01.	£ 84
POWER SCREEN v1.1.	£ 83
Vermont Views v2.0	£ 270
WideAngle v1.0	£ 54

TRANSLATORS

BAS_C v3.2	£ 205
dBx/dBport with Source v4.7	£ 677
FOR_C ++ for DOS	£ 731
FOR_C with Binary Runtime v3.1	£ 367
For_Struct v.1.1	£ 448

UNIX/XENIX

Grammatik	£ 156
Informix 4GL RDS/ID	£ 1869
Interactive ARCHITECH Multi-User Extension	£ 488
Interactive Architech Series Workstation Developer	£ 972
Interactive Motif v1.0	£ 99
Interactive Motif Development System	£ 122
Norton Utilities for System V v3.2	£ 180
Oracle UNIX/Xenix 386 1-user v6.0	£ 1240
RM/COBOL-85 Dvlpt. System v5.0	£ 1046
SCO Microsoft FORTRAN Compiler v3.3	£ 390
SCO Microsoft Pascal Compiler 286 v3.3	£ 373
SCO Open DeskTop v1.1	£ 1150
SCO UNIX System V/386 Development System GT/MC v3.2	£ 535
VEDIT PLUS for UNIX/XENIX v3.24	£ 156

WINDOWS DEVELOPMENT

Asymetrix ToolBook v1.0	£ 232
Btrieve for Windows v5.1	£ 302
Halo Windows Toolkit v1.0	£ 286
hDC Windows Express v3.0	£ 43
KnowledgePro for Windows v1.4	£ 496
Microsoft Windows Device Driver Development Kit v3.0	£ 260
MS Windows Productivity Pack	£ 34
MS Windows Software Dvlprs Kit v3.0.	£ 260
Object Vision v1.0	£ 75
Smalltalk/V-Windows v1.0	£ 227
WindowsMaker v3.01	£ 502

USA SOFTWARE

16 CONNAUGHT STREET • LONDON W2 2AG • UK
PHONE: 071–351 2360 • FAX: 071–352 0574

We will match any advertised Price!

- Prices include 1st Class Recorded Delivery but are exclusive of V.A.T.
- All major credit cards accepted.
- 30 day invoicing for qualified organizations.
- Special discounts for large or corporate buyers.
- Prices subject to change without notice.

UK4/91

Over 3,500 products carried. Call today for price list-disk.

CIRCLE NO. 863

When your system has a weak point . . .

. . . Quintus Prolog is the strength you're looking for.

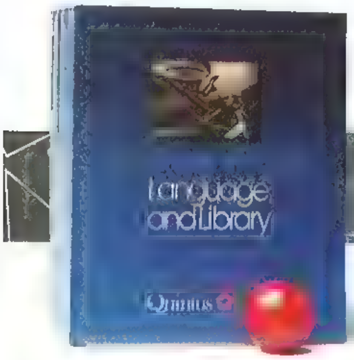
Today's practical systems programmers require robust, powerful and productive software tools and more than just one. No one language offers a complete solution in itself. For example, C is a general purpose programming language offering great portability, whilst Prolog gives enormous programmer productivity for searching databases, manipulating strings, reasoning and matching patterns. Each language is powerful when used individually. When used together their potential is unlimited.

Using each language for what it does best is a good solution to the development efficiency problem. With **Quintus Prolog Release Three** complete integration can be easily achieved using C, C + + and other languages.

You can add power and sophistication to applications without adding reams of code. We can prove it.
Call us, on (0923) 211657.

AII
SEE US ON
STAND 81 AT THE
SOFTWARE TOOLS
EXHIBITION
AII

So avoid weak spots, use the right tools for the job – Quintus and LPA Prolog Products available from A I International Ltd.



AII

ARTIFICIAL INTELLIGENCE INTERNATIONAL LIMITED
Gresham House, 53 Clarendon Road, Watford, Herts. WD1 1LA
Telephone: 0923 211657 Fax: 0923 247836

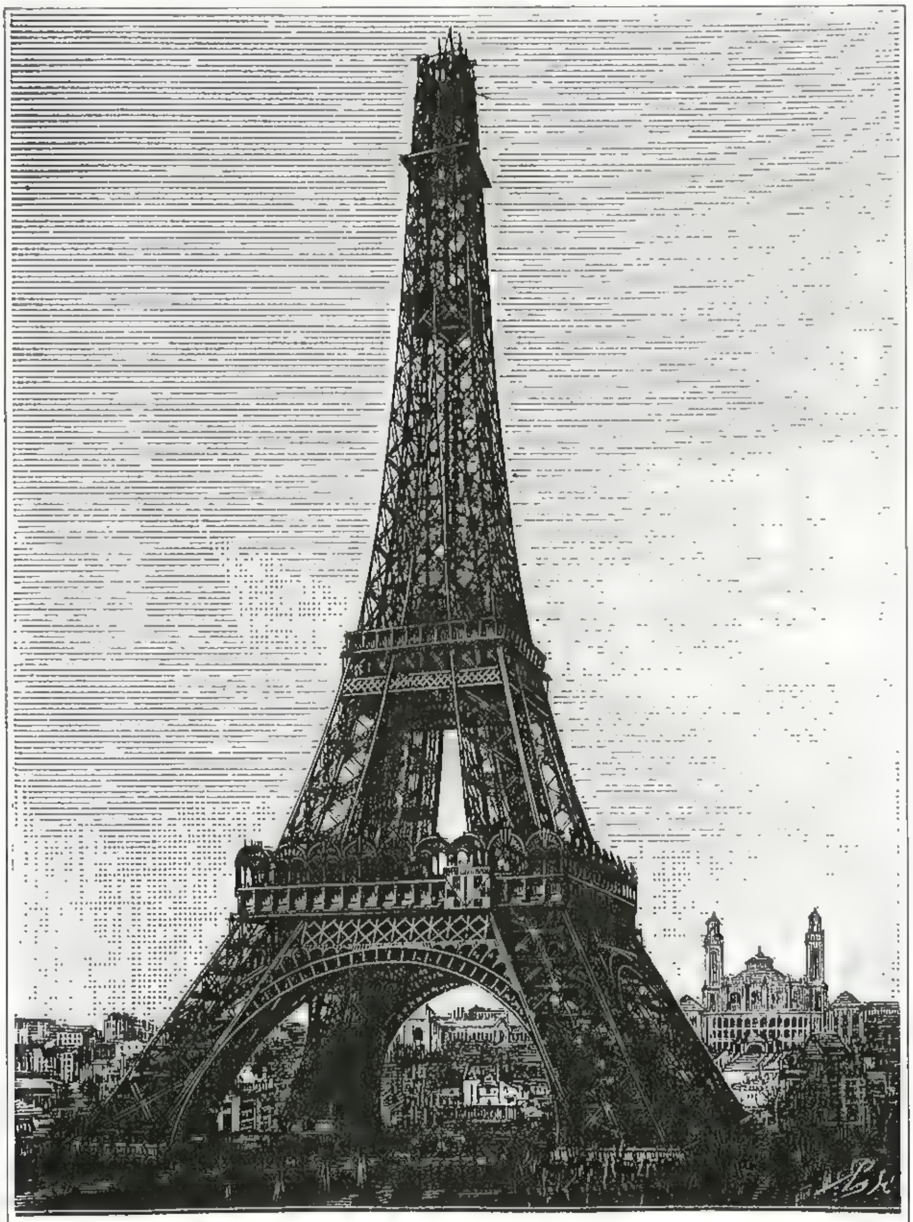
CIRCLE NO. 864

Reuse `implies` Eiffel

Eiffel is designed specifically for producing high-quality, reusable code. Pete Steggles explains why he thinks it's the only tool for the job.

Ask a devotee of object-oriented programming why it's so good, and the first word you'll hear is 'reusability'. But if you ask them how much reusability they get in practice, the answer won't be quite so enthusiastic. The fact is, when you start trying to do some serious software reuse, you run into problems the pundits didn't tell you about. Object orientation, although essential for reuse, just isn't enough on its own.

When Bertrand Meyer designed Eiffel, he had already run into these problems, and he set out specifically to solve them with Eiffel. The best way is to explain what these problems are, and how he succeeded in solving them, to explain a bit about Eiffel as a relatively 'pure' object-oriented language, then look at the problems, and then the solutions.



THE EIFFEL TOWER, IN PROGRESS.

THE ILLUSTRATED LONDON NEWS

Eiffel Basics

To write a program in Eiffel, you write class definitions. In Eiffel, as in all object-oriented languages, a class definition describes the structure of each object which is a member of that class. The structure of an object is defined by the set of 'features' which it contains. There are two kinds of feature: attributes and functions. Attributes are variables held locally by individual objects (equivalent to instance data in other languages), and functions are shared by all the objects which contain them. At run-time, for any given class, any number of objects in that class could be created, each with its own data space but sharing the same function space. Each of these objects can be treated as an instance of a type, and assigned to attributes of other objects provided the assignment is type-correct.

Features can be 'exported' - ie made visible - to other objects. Objects communicate by 'calling' each other's exported features. To call a feature `b` of object `x`, you just write `x.b`, a syntax familiar to C and Pascal users. If `b` is an attribute, then its value is returned to the caller; if `b` is a function then it is executed (possibly changing the values of the attributes in `x`) and its value is returned. Of course, if the value of `x.b` is another object with exported feature `c`, say, then we can make the feature call `x.b.c`, and we can carry on this indirection indefinitely.

The feature call is the only way that objects can communicate; they can't, for example, change the values of each other's variables or call each other's unexported features. Because of this, an Eiffel class is often described as an 'abstract data type', that is a type which has certain defined operations (its exported features) and a hidden mechanism which implements the operations. The object which calls a feature is called the *client* and the object which owns the called feature is called the *server*.

```
-- The 'require' clauses are preconditions;
-- the 'ensure' clauses are postconditions.

class RING_MENU [T] export

   is_empty, is_active, has_title, has_banner,
   can_fail,

   user_answer, new_entry, set_title,
   set_banner, hard remove, fail, key

inherit

   MENU_LOGIC [T]
                    rename   remove as hard_remove
                    redefine remove, key_up_action,
                             key_down_action, options

feature

   height : INTEGER is 4;
      -- Height of top bar including banner window

   overflow : STRING is " ... ";
      -- The overflow indicator.

   banner : STRING;
      -- Attribute which contains the banner.

   set_banner (s : STRING) is
      -- Set the main banner to 's'.
   require
      not is_active
   do
      banner := s
   ensure
      has_banner
   end ;

   options : SEARCH_LIST [RING_MENU_ENTRY [T]];
      -- The list of options for the menu.

               more definitions

   user_answer : T is
      -- Return a single value from the list of options,
      -- determined by a dialogue with the user.
   require
      not is_empty ;
      has_title ;
      has_banner
   do
      Result := single_user_answer
   ensure
      is_active ;
      can_fail = old can_fail ;
      Result.Void implies can_fail
   end ;

   Create is
      -- Handle setting up of data structures.
   do
      options.Create
   ensure
      not can_fail ;
      not is_active ;
      not has_title ;
      not has_banner ;
      is_empty
   end

end -- class RING_MENU
```

Figure 1 - Eiffel code for a generic ring menu class

In each Eiffel class definition there are three main clauses: a feature definition clause, an export clause, and an inheritance clause. The feature definition clause allows explicit definition of functions and attributes. Functions are defined in a simple programming language which has basic facilities like conditionals and loops. The language has a powerful type checking mechanism, of which more later. The export clause lists which of the features are available for calling by other objects.

The inheritance clause allows the programmer to declare all the parent classes from which the new class is to inherit. This new class gets all the features owned by its parents. When inheriting, the programmer can rename and redefine any inherited feature. This inheritance/redefinition facility is, of course, the basic requirement for a language which permits any significant degree of reuse. Renaming is useful for two things: to get round accidental name clashes which might occur when inheriting from more than one class at a time; and to enable the programmer to make the names of features in an inheriting class more in tune with what they are attempting to portray.

All this may seem a mouthful, but the basic concepts in Eiffel are very simple: classes define sets of objects which have features, some of which are exported, some of which aren't; objects communicate by calling other objects' exported features; and objects can come to possess possibly modified versions of other objects by using inheritance with renaming and redefinition.

Fantasy

People who talk about software reuse tend to have at least three different situations in mind.

The first is that of extending a set of services by using inheritance and dynamic binding. For example, in a system I may have a class which provides an interface to some database system from vendor A. I want to provide an interface to vendor B as well (vendor A and vendor B are notorious for the incompatibility of their databases). The way to do this using inheritance is to define a new class which inherits from the old interface class and redefine the innards so that the class interfaces to B's product rather than A's. Then I can just assign an instance of my new class to the client object, who need not know (and quite rightly does not want to know) whether he is interfacing with an A or a B.

The second situation is the easy one of a client using a class. If I want to use a window in an application, I can look through a class library and find a class `WINDOW` already written. All I have to do is use it.

The third case is that of producing a new class by inheriting from completely different ones. There is no attempt to extend the services of the old classes, the inheritance is just a way of scavenging code that works. Here we will often find renaming useful. For example, if I want a window with a stack of subwindows I might inherit from classes called `WINDOW` and `STACK [WINDOW]` (the bit in brackets is a generic type instantiation; if you don't know what that is, don't worry) but now I want to rename the `push` and `pop` features defined in `STACK` to something like `new_subwindow` and `last_subwindow` so that users don't need to know what a stack is to use my software. Once that's done, though, a new class has been constructed just by sticking others together, with minimal extra programming effort.

Three different but common situations; three nice easy solutions. According to the pundits, every company using an object-oriented language will see these sort of situations all the time: experienced programmers will write difficult classes and novices will be able to use the results easily. Productivity will soar, and a new golden age will dawn.

```
MENU_LOGIC [T]
   NAVIGATION
      SCREEN_HANDLER
         BASIC
            BASIC_ROUT
```

Figure 2 - The ancestors of the ring menu class

Real Life

What is glossed over in these three situations is that, if a class is going to be much use, it will have attributes whose value changes. Moreover, at least some of its functions will only really mean anything when the attributes have certain sets of values, and they will not work if those conditions aren't met. In fact, the class will define a 'state machine', and if attempts are made to use it in the wrong way, it will behave bizarrely and the system will crash.

Now when we look back at the three situations, three questions spring to mind. The database-interface-extender says 'what state changes does the application expect when it calls this feature?' The window-user says 'what state changes am I going to get when I call this feature?' If these questions are not answered correctly, then the system produced will keep crashing, because client objects will keep making feature calls which make no sense to the server objects because the server objects are in the wrong state. If these questions don't get answered then software built using reuse will, in general, not work. So how do the questions get answered?

Here, the optimist will mention *class browsers*. Ignore him. Browsers are often very visually impressive tools and are a good way of getting an idea of the structure of a system, but 'an idea' is not what we're interested in: we want to know exactly what the state machine structure of the class in question is, so that we can ensure that our software won't crash. A browser forces us to look at the code to find out what a class really does, and one thing is for sure: looking at the code isn't going to help. Remember, in any remotely impressive example of reuse it's somebody else's code you're using, and looking at other people's code is rarely an enjoyable or illuminating experience.

The only thing left is documentation. This may seem a good solution, but only to those who have never tried to use a software module written in a traditional language by reading documentation not generated by a technical author. We can't use technical authors to generate all the documentation of reusable code because it costs too much, and if we get the programmer to generate it we run the risk (in fact a near- certainty) that it will be unusable.

But it's worse than that. Consider the window-with-subwindows writer. Having produced a new class with almost no effort, he or she will then have to go through the documentation files for the two old classes,

```
class interface RING_MENU [T] exported features

   is_empty, is_active, can_fail, key, fail,
   hard_remove, has_title, has_banner, set_banner,
   set_title, new_entry, user_answer

feature specification

   is_empty: BOOLEAN
         There are no options to choose from.

   is_active: BOOLEAN
         The menu has been set up in windows and
         can be displayed on the screen.

   can_fail: BOOLEAN
         The dialogue can fail when collecting an answer
         (thus returning Void).

   key: INTEGER
         The last key to be read by the menu. This can be
         examined by a client to give nested menus when
         used in conjunction with 'fail'.

   fail (x: BOOLEAN)
         If 'x' is true the menu returns void if it reads a
         key it doesn't recognise. Otherwise the menu will
         simply read another key and can never return void.
      ensure
         can_fail = x

   hard_remove
         Take the menu off the screen.
      require
         is_active

   has_title: BOOLEAN
         The menu has a small title.

   has_banner: BOOLEAN
         The menu has a banner.

   set_banner (s: STRING)
         Set the main banner in the centre of the menu to 's'.
      require
         not is active
      ensure
         has_banner

   set_title (s: STRING)
         Set the left-hand title of the menu to 's'.
      require
         not is_active
      ensure
         has_title

   new_entry (message, explanation: STRING; hotkey: CHARACTER; means: T)
         Create a new menu entry. The arguments have the following meaning:
         'message' is the actual highlightable text.
         'explanation' is a short explanation string for the entry.
         (if it's too long it will just get truncated by the system).
         'hotkey' is a hotkey for the entry.
         'means' is the thing returned when the entry is chosen.
      require
         not is_active
      ensure
         not is_empty

   user_answer: T
         Return a single value from the list of options,
         determined by a dialogue with the user.
      require
         not is_empty;
         has_title;
         has_banner
      ensure
         is active;
         can_fail = old can_fail;
         Result.Void  implies can_fail

   Create
         Handle setting up of data structures.
      ensure
         not can_fail;
         not is_active;
         not has_title;
         not has banner;
         is_empty

end interface class RING_MENU
```

Figure 3 - Automatically generated documentation on the ring menu

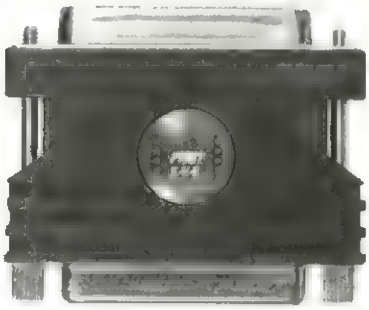
5 out of 5 hackers prefer other software protection methods to Hardlock E-Y-E®

What hackers dislike...
Hardlock E-Y-E was designed using cryptographic principles. It took the experience and know-how of Germany's No. 1 in software protection and the leading edge technology of a US semiconductor company to create the ultimate software protection tool. Hardlock E-Y-E is based on a custom chip featuring secure algorithmic response rather than simple bit swapping or counting schemes.

What software developers like...
Hardlock E-Y-E combines all the features software developers require in a single product: algorithmic response to provide security and an optional non-volatile memory to allow custom configurations. FAST Electronic has made implementation of Hardlock E-Y-E in your software easy. Use HL-Crypt to protect .EXE or .COM files, or incorporate high level language interface routines in your software. The algorithm parameters and the contents of the memory can be programmed in seconds using our Crypto-Programmer card. This unique card guarantees that no one else can burn your original codes. Simply plug the card into any PC slot and start up your own Hardlock E-Y-E workshop.

What your customers will like...
Hardlock E-Y-E allows unlimited backup copies. The device is shipped with the software for the user simply to plug into the parallel interface and forget. Daisy chainability, outstanding reliability (no battery is needed), and the most compact High-Tech design ensure that your customer will accept Hardlock E-Y-E.

What your accountant will like...
Hardlock E-Y-E needs no factory coding. This ensures optimum delivery schedules and stock flexibility. Revenues will go up as software piracy and multiple usage are prevented. Despite its wealth of features, Hardlock E-Y-E's prices remain competitive.



Hardlock E-Y-E programmable, algorithmic response and memory option – all in one.

...As more and more software developers, customers and accountants appreciate the Hardlock E-Y-E device, hackers like it less and less.

MAGNIFEYE

Order your demo unit today. Contact Magnifeye,
235-239 Walmer Road, Walmer Studio #6, W11 4EY, Telephone 071 221 8024, Fax 071 792 3449.

CIRCLE NO. 868

putting them together and changing all the bits which are no longer true because of renaming or redefinition. The very 'mix and match' approach which we are trying to encourage will result in a daunting period of text editing, with errors almost certain.

You think it can't get worse? It can! Think how we usually learn how to use a piece of software: when we're learning it we make errors and we rely on good error-reporting in the software to tell us what we've done wrong. In fact, if the error-reporting is bad, or not there, then we give up using the product and throw it away. Now a class definition is a piece of software like any other, and if we are to use its facilities as a client, then we're going to need good error-reporting. So in future, if we want to have significant reuse, programmers will have to put good-quality error reporting code into all their class definitions.

So, to implement reuse in a normal object-oriented language, each class needs high quality documentation and a sophisticated error-reporting mechanism. This takes a long time to produce. When we use the power of multiple inheritance, the time taken to get the documentation and error-handling right is going to greatly outweigh the time taken to write the code. Ironically, trying to implement reuse with an inadequate language could easily reduce productivity; it could be more effective to throw most code away.

Eiffel to the rescue

These were the problems Meyer was faced with. He was fortunate in having an understanding of formal methods of software development, and in seeing how to apply a simple idea, originally developed in languages like VDM and Z, to solve all the problems at once.

Meyer's solution is to recognise that classes define state machines. He provides an assertion mechanism which lets us say exactly when a function call can be performed and, if it can, what its execution does to the state. It does this by allowing us to define, for each function, *preconditions* and *postconditions*. These are two sets of assertions about the state of the object whose function is being called. The preconditions are the assertions which must be true for the function to mean anything, the postconditions the assertions which the function guarantees to be true when it has terminated. Additionally, Eiffel provides a way of defining 'class invariants', assertions which are always true of the class which possesses them.

What does this buy us? On its own, not a lot. But, combined with a few tools, everything we need.

Two tools provided with the Eiffel environment allow us to extract the appropriate assertions to provide automatically- generated documentation for a class in terms of pre- and post-conditions and invariants. The program `flat` produces a flat version of a class defined using inheritance and, taking care of all the renaming and redefinition which may have occurred, shows it as it would appear if it had been programmed without inheritance. `short` takes a class definition and strips out all the imperative code, all the non-exported function definitions and all the assertions involving non-exported features to give a description of the abstract data type which the class characterises. If assertions are used well, the combination of flat and short can be used to produce high-quality documentation for a class. In particular, with `flat` we have achieved the goal of inheriting documentation while automatically accounting for renaming and redefinition.

What's more, the documentation is checkable. The compiler provides options for checking that the assertions are true when they ought to be, ie when a function is called its precondition is checked (and so is the class invariant) and when it exits, the postcondition is checked (and the class invariant again). If an assertion is violated, then the Eiffel system generates a detailed and informative error message describing which assertion was violated and printing the feature call stack as it was when the assertion was violated.

Figure 1 shows part of the code for a ring menu (a sort of menu which 4GLs tend to provide). From Figure 2 we can see that the code for it is inherited from a number of other more abstract classes, and, as a result,

the documentation relevant to the ring menu is scattered around the class definitions of its ancestors, frequently with out-of-date names. The Eiffel documentation tools pull out the relevant documentation, rename it appropriately, and present it in an elegant way (there are options available on the tools to typeset the output they generate).

Figure 3 shows documentation for the ring menu class constructed entirely using `flat` and `short`. Suppose we want to use a ring menu to get a choice from the user. We see that we want to execute the function `user_answer`. By looking at the preconditions of this function, we see that we need to ensure that: the menu has some entries in it; the menu has a title; and the menu has a big banner title. By looking at the postconditions of other routines we see that we can ensure the correct working of the menu by calling `new_entry`, `set_title` and `set_banner`. Should we go wrong, the system will tell us which assertion we violated.

It takes a while to realise just how much we gain from assertion checking, and the `flat` and `short` programs. By including the appropriate assertions we get a test suite, error handler and inheritable, checkable documentation for free.

The Icing on the Cake

There is not enough space to describe the other benefits of the Eiffel language: so we'll have to forget the strong type system, genericity, expanded types, generalised iterator classes and switchable garbage collection. Suffice it to say that Eiffel isn't just a solution to one problem, but is a serious software engineering language.

The Eiffel environment provides more than just flat and short, of course. Included in the package are: an incremental compiler which calculates file dependencies itself; an X-based graphical browser and a browser for use on dumb terminals; a debugger and code viewer, which you won't need to use very often, and a library of over 200 reusable classes, documented automatically with flat, short and ancestors, providing generic structures like lists, trees, hash tables etc; parser generators; X and dumb terminal windowing facilities; and support for persistency. Eiffel also interfaces easily to C (as it should, the compiler produces C output) and has no run-time license fees.

Major research work is being put into developing Eiffel for the future. The EEC Esprit project 'Business Class' is devoted to developing tools for designing systems for implementation in Eiffel, and libraries of classes for building business systems in Eiffel.

Eiffel's basis on solid theory opens up more prospects as time goes by. For example, there is a lot that can be said in a formal way about the reusability of classes just by looking at the state machines which they represent. In the near future, expect to see tools which automatically calculate a figure for the reusability of a given Eiffel class, and suggest modifications to enhance reusability. The true 'software factory' is not far away when you use Eiffel.

People who use Eiffel know that this article's title is true. Unbelievers should worry about is its logical consequent (by *modus tollens*): not Eiffel implies not Reuse.

EXE

Pete Steggles works for Applied Logic Research Ltd as team leader on the Esprit Business Class project. ALR's sister company, Applied Logic Distribution (081 780 1088), is the UK distributor of Eiffel.

I THOUGHT THIS WAS QUALITY SOFTWARE-NOW MY SYSTEM'S JUST HANGING!
Don't leave your customers HANGING. . .

BEFORE YOU SHIP ANY PROGRAM, CHECK IT WITH

COMPUTER LANGUAGE PRODUCT EXCELLENCE AWARD 1990 Jolt COLA

BOUNDS-CHECKER™

WINNER PC MAGAZINE BEST OF 1990 JANUARY 15, 1991 PAGE 168

FINDS OUT-OF-BOUNDS MEMORY ACCESSES *AUTOMATICALLY*

Your program may have 10,000 to a million lines of code. It may occasionally hang mysteriously or it may appear to run flawlessly every time. But under DOS, how can you ever be sure that your program is not corrupting memory it does not own? The only way to be 100% sure is to BOUNDS-CHECK before you ship.

To use BOUNDS-CHECKER you build your program with debugging information (we support most compilers including Microsoft, Borland & JPI). Then you just type <BC file-name>. BOUNDS-CHECKER sets up the 386™/i486 for protection and lets your program fly. If your program accesses memory it does not own or overwrites its own code, BOUNDS-CHECKER pops up displaying the offending SOURCE-LINE or instruction.

Programming under DOS is a gamble, so why not stack the odds in your favor--CALL TODAY.

(603) 888-2386

Call by 4:00 PM EST TODAY and ask us to EXPRESS you an info packet. In most cases you will receive it by 10:30 AM tomorrow. (USA only)

All Nu-Mega products require a 386, 386SX or 486. MS-DOS and Codeview are trademarks of Microsoft Corp. 386 is a registered trademark of Intel Corp. Nu-Mega, BOUNDS-CHECKER, Soft-ICE and CV/1 are trademarks of Nu-Mega Technologies, Inc.

OTHER FINE NU-MEGA PRODUCTS. . .

The ultimate systems debugger.

Soft-ICE™

Debug:

- Interrupt routines
- Device drivers
- T&SRs
- ROMS
- Applications
- Overlays

Features:

- Break out of a hung program
- Real time Break-Points
- Back-Trace history
- Works with other debuggers

If you are debugging an application, Soft-ICE is seamlessly integrated with BOUNDS-CHECKER so you can easily go back and forth between BOUNDS-CHECKing and debugging: a combination many programming professionals can't live without.

1988 BYTE AWARD OF DISTINCTION

Run CODEVIEW for Windows on a single monitor
CV/1.........$129
• Runs in a window • No annoying flash •
• Runs on any display that supports windows •

™ Nu-Mega TECHNOLOGIES INC

P.O. Box 7780 • Nashua, NH • 03060-7780 U.S.A.
(603) 888-2386 • Fax (603) 888-2465

BOUNDS-CHECKER	$249
Soft-ICE	$386

Special Offer
Buy BC & S-ICE ....Save $100

30 Day
Money Back Guarantee

CIRCLE NO. 873

A New xBase for UNIX

Recital version 7.0 is out this month. It offers dBASE and Clipper programmers a leg up to UNIX with a vengeance. Mark Adams inspects the new version.

Reviewing Recital is difficult. The promotional material bills it as 'dBASE for UNIX or VAX'. If you're an MS-DOS programmer, that may sound attractive, until you learn that it doesn't run on DOS. If you're a VAX programmer, you'll probably use Oracle, Sybase or Rdb, and may well find little support for the idea of using a dBASE clone from your peers, and hence, little real interest in a review of Recital 7.0.

The way Recital is designed also makes it difficult to review in a programmers' magazine which has traditionally shunned 4GLs. Although Recital is a comprehensive programming language, it bills itself as a '4th generation environment' (4GE) and has all the frills of a 4GL. To do it justice, it would have to be reviewed as a 4GL, and compared to Unify or Informix.

This review looks at Recital as a dBASE-like programming language to run on UNIX which will allow a screed of old dBASE programs to be taken into a multi-user environment. This is done first because Clipper programmers frequently ask the question 'How can I do UNIX?', and second because this reviewer has inadequate experience with 4GLs or Rdb.

What is it?

Recital is not a new product. It has had a relatively low profile since it was launched in 1986. Version 6 has been around since 1989, and the new release, Version 7.0, is available from 25th May 1991.

Recital is termed an 'advanced RDBMS and 4GL for VAX and UNIX computers' by its makers. It runs as an interpreted language which is automatically compiled when run. It comes with a data dictionary, mammoth dBASE and Clipper superset commands and functions, forms and screen generators and a report writer with report writing language. It implements the dBASE-like Assistant. Surprisingly for a product running on VAX and UNIX, SQL and client-server models are not really part of this release of Recital. The company says it prefers to get the product more established before it tries for the more ambitious architectures, which will include server links to Oracle, Sybase, Ingres and Informix databases.

A PC Server is part of the package, and a Library of access routines for C, FORTRAN and COBOL is provided, as are interfaces to RMS, Rd and other applications software.

Recital comes in any number of disk or tape formats, suited to the environments it runs in: everything from 386 SCO Xenix 2.3.3 to DEC RISC VAXes on Ultrix 4.0. It takes in the Acorn R140, Bull, DG, IBM RS/6000, NCR, Harris, Sun, NCR, Sequent, Prime and Unisys machines on the way. The review was carried out on a 20 MHz 80386 PC with 4 MB of memory, running Interactive UNIX version 2.0.2.

There are three manuals covering commands and functions, a manual on using Recital with the Assistant, a library reference manual and a guide to the Recital environment. They are bound, not ring-bound, and resist all attempts to lay them flat. They have more than a reasonable share of typos, but I'm advised by Recital that new versions will be out in two months. The software came on seven 1.2 MB floppy disks, with an additional Install disk and a PCServer disk.

Moving up from DOS

The claim of compatibility is one that Recital's manufacturers take very seriously, and it's worth seeing how the product stands up to testing on this. The commands and functions in Recital are a real superset of previous xBase languages. They are comprehensive, thorough and detailed. I might even say imaginative. It's difficult to find specific Clipper, Fox or dBASE syntax which falls down under Recital, though a summary of SET commands and functions common (and missing) in both is shown in the separate box.

In practice, moving a trial application system from FoxBASE+ on SCO Xenix to Recital on Interactive UNIX was not quite plain sailing. A system comprising five procedure (.PRG) files, each of around 50 separate programs totalling 582 KB of source code was fully converted in two days. Typical errors were like the `sys(30)` function (returns user's logged-in name) used in Fox but not supported in Recital. An application which relies on this to (say) create user-specific databases will need rewriting, and a kludge involving setting up a memory variable with the user's ID at operating system level before starting Recital and then using `GETENV()` to read this variable works but is messy. The Recital approach to screen drawing allows terminals to be used relatively easily. But it also means that it is badly suited to using traditional `@ SAY...GET` screens. Developers are advised to use Recital's MENU commands instead, and this is something I would also recommend, though the re-coding can be extensive. There are other small problems - EDIT will not work if used with a specific record number, for example - but they are minor. No extensive recoding was required, and most changes were, to be honest, to overcome original design and coding shortcomings.

```
procedure totaliser
m total=m 1+m 2
@ 15,40 say m_total
return

set postrecord to totaliser

@ 10,40 get m 1 postfield totaliser
@ 11,40 get m 2 postfield totaliser
```

Figure 1 - Demonstrating Triggers

Windows 3.0 DLL
Not Included!
ROYALTY FREE

Objective dBASE programmers try C++

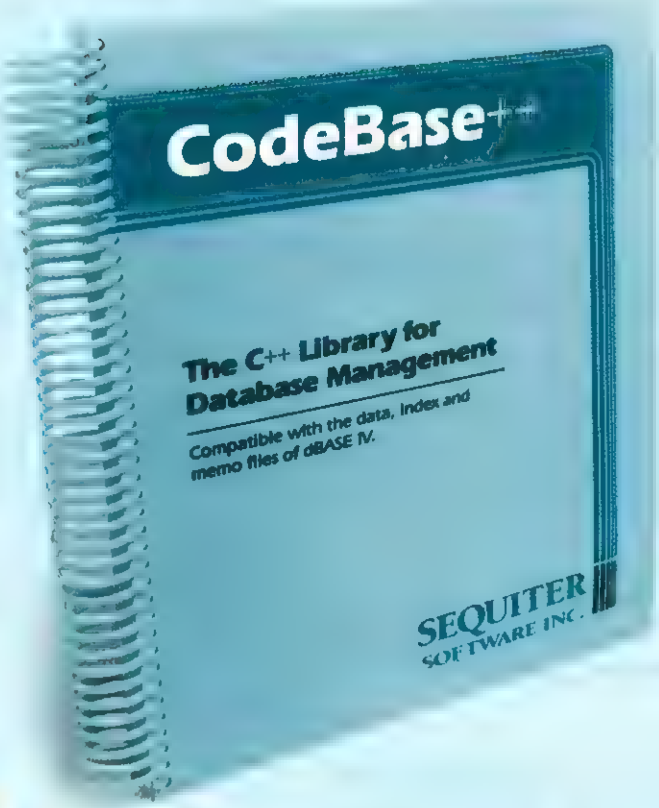
Use CodeBase++ with C++ to build multi-user, dBASE compatible programs. Take advantage of object orientated programming to quickly create maintainable, readable software.

dBASE Compatible

Use CodeBase++ classes to access and change the data, index and memo files of dBASE IV and III. Use other classes to manipulate strings, sort at high speed, perform date arithmetic, evaluate dBASE expressions, manipulate linked lists and manage memory.

Portable

Port your programs between DOS, Microsoft Windows, and OS/2. Keep all the profits as you distribute your programs royalty free.


CodeBase++
The C++ Library for Database Management
Compatible with the data, index and memo files of dBASE IV.
SEQUITER SOFTWARE INC.

Fast

Watch as your C++ programs execute much faster than corresponding dBASE, Clipper or FoxPro programs. Be surprised at the small executables and low memory requirements. Examine the C++ source code, which is completely included, to learn good object orientated programming techniques.

Order Today

Order today at £195 by contacting your nearest software dealer. Discover why Sequiter Software Inc. and participating dealers offer a 60-day money back guarantee.

SEQUITER
SOFTWARE INC.

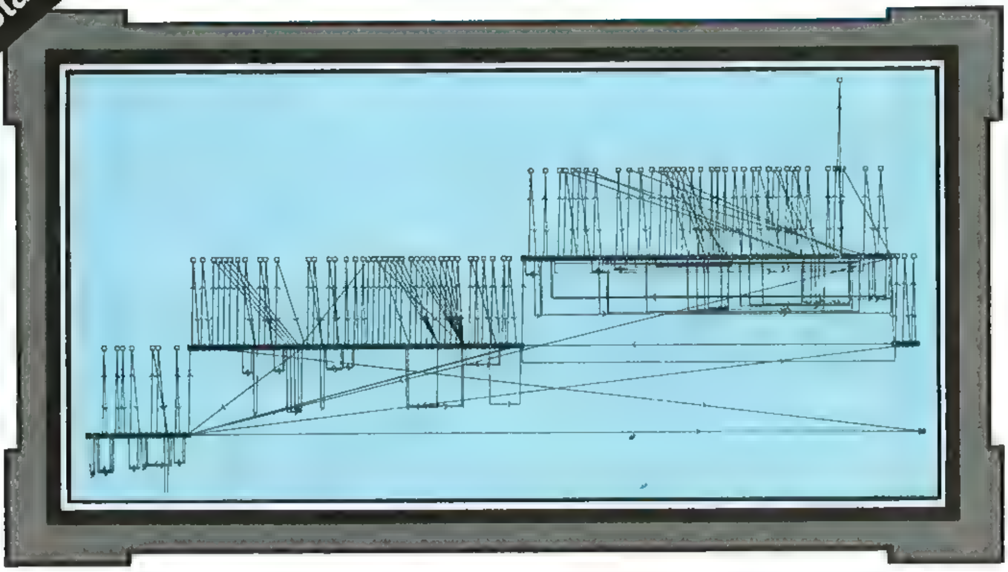
Call 1-403-448-0313
Fax 1-403-448-0315
Hours: 14:30 - 24:00 GMT

P.O. Box 5659, Station L, Edmonton, Alberta, Canada T6C 4G1

CIRCLE NO. 874

See us at
SOFTWARE TOOLS
on stand 138

ART or RE-ENGINEERING?



A control flow graph from

LOGISCOPE

◆ This could be an example of post-modernist art by a famous painter. Unfortunately, it is LOGISCOPE'S graphics analysis of some very poor source code.

◆ To help you get the picture on your software quality VERILOG developed LOGISCOPE, a powerful software analysis and testing tool.

◆ LOGISCOPE provides you with a concise view of the structure of your source code. It allows you to see a picture of the decision paths that run through each module and by using VERILOG's Call Graph a view of how they are connected is easily obtained.

◆ The quality of your code can be analysed and measured by using a comprehensive list of software metrics generated by LOGISCOPE.

For further information about LOGISCOPE or any other VERILOG case tools, please write to Bernadette GLOVER, VERILOG UK Ltd, 34 The Quadrant, Richmond, SURREY TW9 1DN or ring us on 081 940 2212, or fax us on 081 940 2933.

◆ LOGISCOPE can help you clearly define you test programs and provide invaluable information on their effectiveness with coverage rate results and analysis.

◆ LOGISCOPE can help you clearly define your test programs and provide invaluable information on their effectiveness with coverage rate results and analysis.

◆ LOGISCOPE is available for most common languages including: Pascal, C, Cobol, Fortran, Modula 2, PLM, Assembler and Ada.

VERILOG's team of re-engineering professionals are ready to help you implement a case solution to solve the toughest of re-engineering challenges.

CIRCLE NO. 875

VERILOG

Using a 4GL!

In many ways, Recital discourages 'programming' and encourages '4GLing'. It seems an utter waste of time to hand code valid clauses, help messages, @ SAY and @ GETs, and menus with the sheer numbers of options for each standard command available. A glance at the separate box will illustrate this. @ GET alone can include up to 20 keywords (eg CALCULATED BY, CALCULATE, COLOR, CHOICELIST, PRERECORD, POSTRECORD, ERROR, FUNCTION, PICTURE, MUST_ENTER, RELATION INTO).

Many developers shy away from menus, screen forms and views in programming. They often want to 'dip in' to generated code to modify or tweak to their own preferences. I can't find any justification for sustaining these practices with Recital. CREATE SCREEN, for example, is so powerful, you'd be wasting your time to hand code. The screen painter is pretty flexible, and gives immediate access to triggers and all data dictionary attributes. It is possible to build really thorough code complete with full colour control, input validation, complex views and relations to other databases and access to help files using just CREATE SCREEN. It even extends to letting you write or modify your trigger programs without leaving the screen editor. Nice. CREATE SCREEN is the most important part of the 4GL, but the report writer incorporates the same principles.

Having said all this, Recital will still find great affection with 'real' programmers. Its syntax will permit complete hand coding of good looking applications with statements and commands broadly similar to Clipper.

The designers of Recital 7.0 placed a healthy emphasis on devising programmers' tools to create easy-to-use applications which also provide data integrity. The ultimate aim of developers, I think, should be to present users with data entry screens on which is it impossible to enter invalid data, yet where at all times, the user is offered guidance on how to enter correct values. Recital offers ample variants on @ GET, such as CHOICELIST, to achieve this. Recital also takes this one step further with a data dictionary. Ultimately, the data dictionary makes it virtually impossible to enter invalid data, and yet makes it easy to help users enter valid data.

```
* MASTER contains customers, with key fields
* sn (surname), co (company) and recnum (record number)
* ORDERS contains details of orders placed by
* customers in MASTER. ORDERS and MASTER are
* linked using the RECNUM field.
* There may be up to 10 orders per customer.
* Databases are set up as normal using SET RELATION

select 1
use orders index orde_rec
select 2
use master index mast_rec
set relation to recnum into orders

* Now table field is defined.

define table ORDER_SUM for ORDERS;
       as 10 rows relating RECNUM by MASTER->RECNUM
go top

* Prepare screen display

@ 6,9 say "DATE    AMOUNT       GOODS            RECORD"

* Now MASTER is SKIPped through, and for each customer
* all previous orders are displayed, and may be modified
* using GET

do while .not. eof()
    @ 2,9 say "Record Number: "+master->recnum picture "@B"
    @ 3,9 say "          Name: "+master->sn
    @ 4,9 say "       Company: "+master->co

    @ 8,9 get order_sum!orders->invdate
    @ 8,24 get order_sum!orders->invamount
    @ 8,44 get order_sum!orders->item1
    @ 8,60 get order_sum!orders->smrecnum
    read
    skip
enddo
```

Figure 2 - Use of the TABLE field

The data dictionary encourages programmers to make all transactions directly with database fields rather than with memory variables. When using CREATE [DATABASE], each field can be defined not just as a data type, but with a whole list of other parameters such as a PICTURE clause, VALID clause, error message, choices in the form of a choicelist, range, help messages, trigger procedures, calculation parameters if required, a MUST_ENTER option, default value and a recalculation option.

The Data Dictionary is not a new component of Recital, but version 7.0 has been updated with new triggers (see below for explanation of triggers) at the database level. Triggers are now offered on UPDATE, DELETE, APPEND, OPEN, CLOSE and ROLLBACK commands so that if any of those commands are encountered a specific PRG can be executed. It adds a valuable extra level of security to any application.

For my part, there is a big downside to the dictionary. Each dictionary remains specific to a database, rather than to a system. So when one database is created with a number of fields which also appear in other databases, the dictionary for the new database has to be fully specified from scratch. If the dictionary is used fully (which I recommend), each field could have up to 30 separate items specified which have already been specified in previous databases.

Triggers and Tables

Triggers are commonplace in larger databases, but not so common lower down the scale, and are well worth getting to know. A trigger is a piece of code placed in a procedure file which is executed when a specific event happens. Triggering events may be when a database is USEd, when a record is entered with a GET, or when a new record is accessed for reading.

A typical trigger may be to total fields on a screen of GETs as individual GETs are entered. This type of validation is possible, and can be reasonably efficient, using VALID clauses on older systems and lPreCondition and lPostCondition on @ GETs with Clipper 5.0. But using VALIDs in this way stretches them too far and lPreCondition and lPostCondition are overly basic to do anything meaty. Recital would do it by specifying a procedure to total record values and by specifying a POSTRECORD trigger which called this procedure, as shown in Figure 1.

Equally, a PRERECORD trigger could be specified, and PREFORM and POSTFORM triggers can be used on full record EDITs, CHANGEs or with forms. Recital 7.0 gives the option of triggering on entry and exit from a field, entry and exit from a record, entry and exit from a form and the database-level triggers listed above.

Another feature new to Recital 7.0 is the TABLE field, which everyone will love. If you have a system where a single record like a customer is related to a number of other records such as orders in an orders database, you will have found a need to display all orders for a given customer at one time. It's a tricky business usually requiring loops of LOCATEs and CONTINUEs if you're lucky or DO..WHILE..SKIP if you're unlucky. Even if it works, it's difficult to get it all looking nice.

Tables fields solve all this. They're a cinch to program, quick in practice and can look really professional. It's easiest to explain by reference to Figure 2. It looks brilliant and works well apart from one major problem. It is not possible to use @ SAY to display the orders information. The table field insists on having @ GETs, which is often not what is required.

Taking Care of Things

Recital seems particularly good at providing features to deal with the complex task of setting up systems on multi-user machines. Given that Recital runs on VAXes, this is an absolute prerequisite. Memory caching is offered for indexes and databases, and the MENU commands, reduce looping and screen painting, achieving savings in screen drawing and processor use.

```
@ GET [CALCULATED BY]
[CALCULATE] [CHOICELIST @<alias>,<exp>]
[HELP] [LOOKUP IN <alias>] [MUST_ENTER] [NOECHO]
[READ_ONLY [IF <condition>]] [RELATION [INTO <alias>]]
[VALID IN ] [VALIDATE WITH <prog/proc>]
```

Figure 3 - The Comprehensive GET command

Recital commands are also generally pretty comprehensive, @ GET, for example, includes options which I can't seem to find on any other comparable PC database language (Figure 3).

Typical UNIXy problems, such as PC keyboard mapping, are sorted immediately by the provision of PCKEYS and PCGRAPHICS commands which bring quick compatibility with PC AT systems and welcome relief.

There are a few niggles with Recital 7.0. The CREATE SCREEN command cannot access dbfs located in other than the current directory. CREATE SCREEN does not support block moves of characters or fields. Help files and the manuals contain typing errors. Help is really comprehensive, but not context sensitive, and can require seven separate menus to reach the desired item. Pull-down menus on Recital CREATE screens do not allow left-right scrolling when an option is selected. Screen drawing on a PC is slow, almost terminal-like in performance. PACKing an empty database with SET PERFMETER ON causes a core dump. Some of these are bugs and some are design problems.

As at the time of writing, Recital has plans for version 7.1 at the end of July, which, the technical department tells me, will address the bugs listed.

Conclusions

It has not been possible to benchmark Recital 7.0, primarily due to the complexities of ensuring that UNIX systems are equivalently configured and optimised. This review is the weaker for it, and there certainly do remain questions over Recital's screen drawing and procedure files loading when using converted software from dBASE. If applications were written from scratch using Recital's optimised commands, these would be less of a problem.

With a background steeped in the reliable but under-featured, inflexible and monochrome FoxBASE+ and QuickSilver on Xenix, my reaction is probably over the top with enthusiasm for Recital. But it is an exciting product which really lets the creative UNIX programmer free. It is comprehensive, and has been well tested and well designed over five years. Newcomers in the dBASE for UNIX battle, including Ashton-Tate, will find it tough.

Prudence forbids an all-out recommendation to purchase Recital, but if you come from a dBASE background and are moving up, it would be foolhardy for you not to evaluate it for yourself.

Mark Adams has written several large xBase systems, and toiled long and hard with 'moving up to UNIX'. He is currently writing and programming freelance, and may be contacted at the .EXE offices.

Recital 7.0 costs £2195 for development system two-user licence, £1095 for an equivalent runtime licence. Pricing is the same across all UNIX platforms. Recital Corporation is on 071 401 2727.

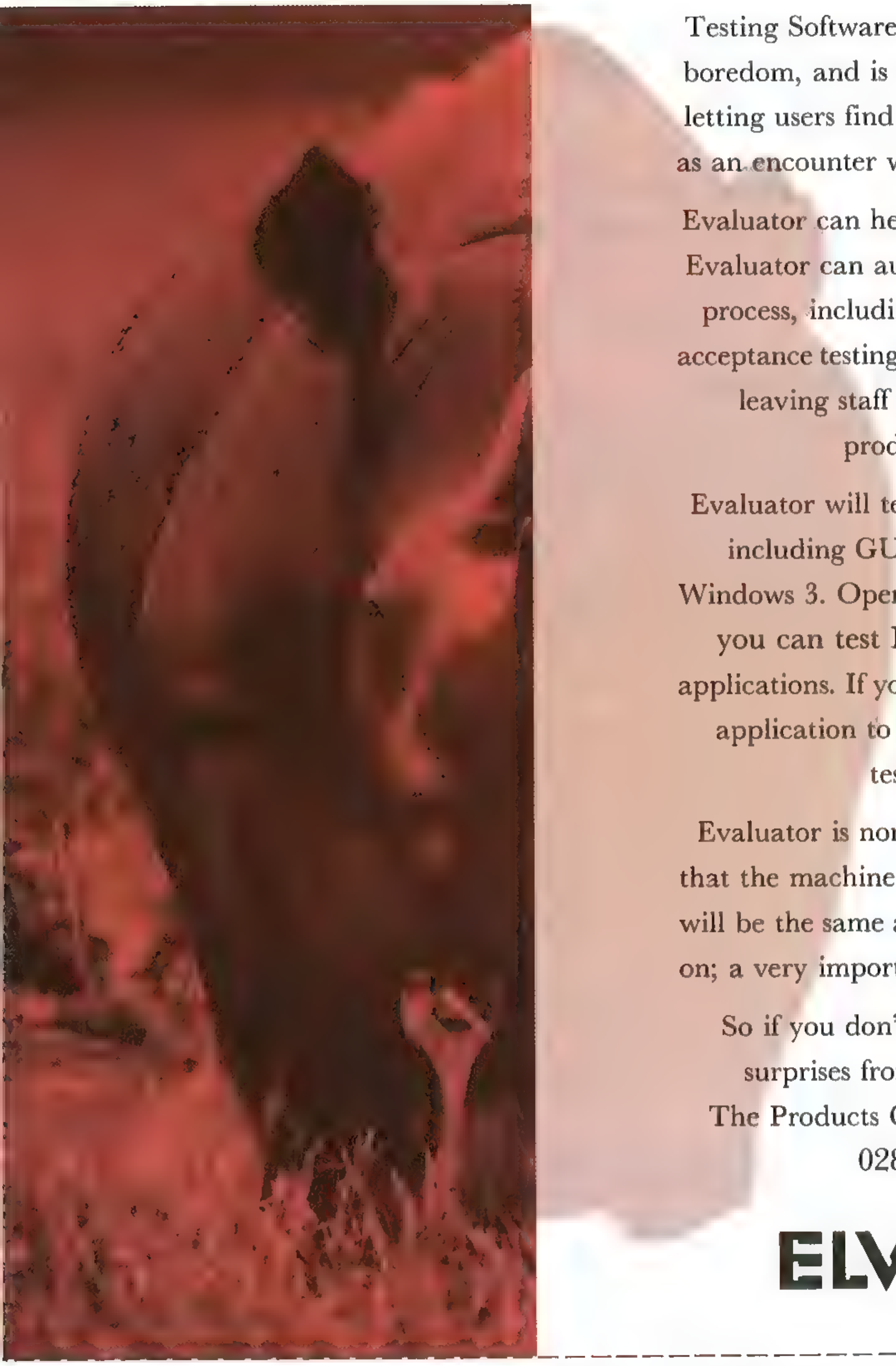
Recital/Clipper 5.0 comparison

There are 148 functions offered by Recital which are not in Clipper 5.0. Of the 148, alternatives to 33 are provided in Clipper, though not with directly compatible syntax. The remaining 115 functions in Recital, which are not in Clipper 5.0 include 6 basic trigonometric functions, some obvious UDFs like PAYMENT() and PERCENT(), graphics functions and relatively unimportant string manipulation functions as well as network and tuning functions. The majority of the remaining functions tend to offer answers to 'where am I?' questions: ORDER(), LINENO(), PROMPT(), NETWORK(), TAG(), COMPLETED() and CAPSLOCK(). Good programmers will argue that the system should always know these things and shouldn't have to refer outside the program.

There are 144 SET commands offered in Recital 7.0 which are not offered in Clipper 5.0. Of this 144, there are 30 which have alternative, but not directly compatible syntax in Clipper 5.0. Of the remaining 114, the majority are for help management, performance tuning, windows and memo handling, trigger handling, graphics attributes, time and debugging.

There is 1 SET command, SET EPOCH TO, in Clipper 5.0 and not in Recital 7.0. There are 17 functions in Clipper 5.0 which are not in Recital 7.0 including WORD(), VALTYPE(), DBSTRUCT() and BBROWSE().

Software errors can be brutal...
...find them first with Evaluator



Testing Software properly means hours of boredom, and is very, very expensive. But letting users find errors can be as bruising as an encounter with our friend on the left.

Evaluator can help. A unique testing tool, Evaluator can automate the entire testing process, including regression, stress and acceptance testing. It can do this unattended, leaving staff free to perform more productive work.

Evaluator will test every kind of software including GUI applications such as Windows 3. Operating system independent you can test DOS, OS/2 or UNIX applications. If you can get your mainframe application to display on a PC it can test that too!

Evaluator is non-intrusive, which means that the machine you test your software on will be the same as the machine it gets used on; a very important quality consideration.

So if you don't want any unpleasant surprises from your software, call The Products Group for full details on 0285 - 655888

ELVEREX

QA

Tick here for full details:

CIRCLE NO. 876

□ Evaluator
□ Product Catalogue
□ Consultancy
□ Training: OS/2, UNIX, Windows, C, Object Oriented, Networks.

NAME TITLE

COMPANY

ADDRESS

TEL. NO.

EXE 6/91

Mail to: The Products Group, QA Training Ltd., Cecily Hill Castle, Cirencester, Gloucestershire GL7 2EF. UK.
Tel: 0285 655888 Fax: 0285 640181

Know your FPUs

Does your software ever detect a ghost FPU and crash? Avoid trouble with Bob Stimpson's comprehensive set of detection routines.

First was Intel. Then came Weitek. Now AMD, Cyrix and Integrated Information Technologies (IIT) have joined the pack. All provide PC Floating Point Units (FPUs) or maths co-processors. FPUs fall into two families - the *peripheral* and the *memory-mapped* types. This article concentrates on peripheral FPUs - the Intel 80x87 family and compatibles. Lack of space prevents the presentation of detention routines for memory-mapped FPUs.

Differences between the Intel compatibles are subtle, but can prove useful from the programming point of view as well as from the obvious benefit of enhanced number crunching speed. These are outlined and the detection routines explained, along with a brief performance comparison between the Intel, Cyrix and IIT offerings.

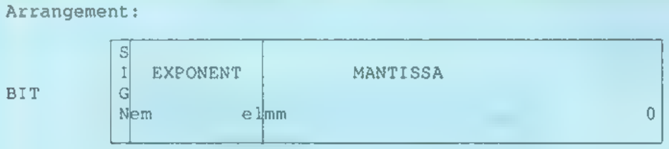
Standards

The standard to which all current FPUs (and most current compilers) store and process real numbers is the IEEE 754 (1985) *Standard for Binary Floating Point Arithmetic* specification. Figure 1 shows the formats for IEEE real and BCD numbers. The 80387, introduced in 1987, required an extended instruction set when compared with the earlier 8087 because the latter only met the draft edition of the standard.

Peripherals

Intel 80x87 peripheral FPUs sit on the CPU periphery using a dedicated data I/O bus. The generic 8087/80287 coding standard is the lowest common denominator, providing the widest portability of code. The 80287 provides additional protected mode functions over the 8087; the 80387 adds Comparison, Remainder, Sine and Cosine functions. The i486 coprocessor extension is simply both CPU and FPU cast in the same VLSI die. The 487SX is the i486 FPU in a separate chip.

The Intel standard provides users with eight 80-bit registers and a comprehensive set of operating instructions acting in stack type operations. All data is converted upon loading to an 80-bit *Temporary Real* format which the FPU uses in all calculations. The range of this internal type ensures all the external stored 64-bits are valid. The FPU converts to the destination format automatically whenever a value is stored back in memory. A benefit of this is that an IEEE format real can be converted to a 2's compliment integer simply by loading a Real (FLD) and saving an Integer (FIST) - taking around 160 clock cycles in all. This uses the rounding mode defined in the FPU Control Word.



REAL DATA TYPES:	Size: BYTES	Bit Locations in Type: MANTISSA mm	EXPONENT el	em	SIGN s
Single Precision	4	22	23	30	31
Double Precision	8	51	52	62	63
Extended Precision	10	63	64	78	79
BCD:	10	71	72*	78*	79

*The exponent field is unused and ignored by the FPU for BCDs.
Integers can be Words, Double-Words or QuadWords in 2's compliment format.

Figure 1 - Format of IEEE Real Data Types

2a - Detecting presence of a FPU using the BIOS.

```
xor ax,ax       ;clear ax
int 11
and ax,2        ;check bit 1
cmp ax,0        ;is it 0?
jnz FPUHere     ;if not, FPU Found
```

2b - Detecting Intel presence using FPU code.

```
CWord EQU word ptr [sp-2]

Push bp       ;set up stack pointer
mov bp,sp
sub sp,2

FWAIT         ;insert wait for 8087
FNINIT        ;initialise coprocessor

FWAIT         ;insert waits for 8087
FNSTCW CWord     ;store to
                 ;control word
CMP BYTE PTR CWord-1,03h ;upper byte
                         ;03h?
JNE fc exit ;if not, no copro present
```

Figure 2 - Detecting presence of an FPU

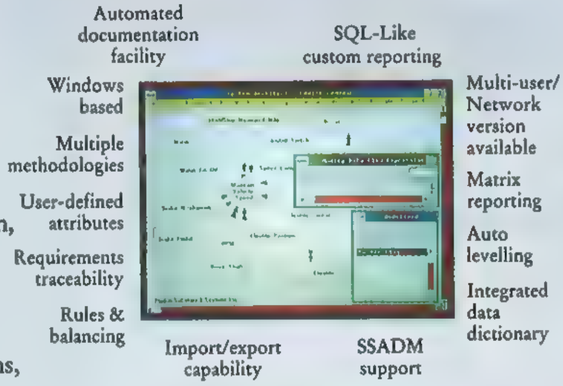
All CPU to FPU data transfers must be made via memory - the FPU cannot access CPU registers directly. The only exception to this is the saving of FPU Control Word to the CPU AX register on 287 and 387 FPUs.

To ensure the CPU and an 8087 FPU remain synchronised, the `FWAIT` command must occur before all FPU instructions. From the 80287 onwards, hardware synchronisation was introduced, and it is only necessary to use the `FWAIT` command between FPU and CPU instructions to ensure the CPU and FPU do not read or write to the same address in the wrong order. The MASM assembler directives `.286`, `.287`, `.386` and `.387` each disable the generation of the default 8087 `FWAIT` instructions in unnecessary positions.

Intel upgraded the 80287 in July 1990 by creating the 80287XL and 80C287A (for portables) which are identical in function, accuracy and instruction set to the 80387.

System Architect is perfect for the beginning CASE user, yet has the power and flexibility to meet the needs of the most experienced users and largest applications.

System Architect has the power and flexibility you need from a CASE product. It works with today's most popular methodologies, including DeMarco/ Yourdon, SSADM, Gane & Sarson, Ward & Mellor (real-time), entity relation diagrams, decomposition diagrams, object oriented design (optional), state transition diagrams, and flow charts. It supports an integrated data dictionary/encyclopedia, and allows multi-user support both with and without a network. What's more, System Architect's open architecture lets you easily import and export data to other products.


Automated documentation facility
Windows based
Multiple methodologies
User-defined attributes
Requirements traceability
Rules & balancing
Import/export capability
SSADM support
SQL-Like custom reporting
Multi-user/ Network version available
Matrix reporting
Auto levelling
Integrated data dictionary

"System Architect stood out from many other prospects because it had the best CORE technology, as demonstrated by it's reliability, flexibility and performance."

Toshiba Corporation

Yet System Architect is more than just powerful. It's easy to use. Microsoft Windows-based and graphics oriented, System Architect lets you get up and running right away. And if you get lost, you can call on a context-sensitive help facility as well as a novice mode.

"It is very easy to learn and simple to use. We use System Architect at the heart of our project environment because it is so flexible in logical and physical terms of the data dictionary."

National and Provincial

At £1,110 System Architect is very affordable. It runs on any PC, and it won't run away with your budget.

"Think productivity has to be pricey? Think again. This product is truly a price performance leader."

System Builder

If you looking for a CASE product with power, ease of use, and affordability, look to System Architect. It's a concept whose time has come.

System Architect™

Supporting IBM's AD/Cycle

TO ORDER
TEL: (0926) 450858
FAX: (0926) 422165

Real-Time Software Limited
118-120 Warwick Street, Royal Leamington Spa, Warwickshire, England CV32 4QY.

Please allow 14 days for delivery.

MICROSOFT. WINDOWS.
Version 3.0 Compatible Product

CIRCLE NO. 877

"With or without a mouse, it's the cat's whiskers."

New development environments and technology transition make 80's editors inefficient and obsolete. To be competitive, programmers must have leading edge tools. The tool you'll use the most, the one that wraps everything into a single consistent environment will be the powerful, configurable SPE Professional Editor.

Right out of the box, it'll make you more productive, more adaptable, more imaginative. Its instant installation, elegant mouse support, advanced user interface, and point-and-shoot help get you running immediately.

MS-DOS, OS/2 and Dual Mode together

MS-DOS, OS/2 and Dual Mode versions are all included in the same box on both 5¼″ and 3½″ disks, with templates for popular languages, with or without a bundled Microsoft® Mouse. Great value, and your team can move into the 90's with the same editor.

Turnkey emulations

If you prefer the commands and keystrokes of other editors, our turnkey emulations duplicate them precisely. And you still gain the SPE Professional Editor's advanced features, windowing capabilities and powerful engine.

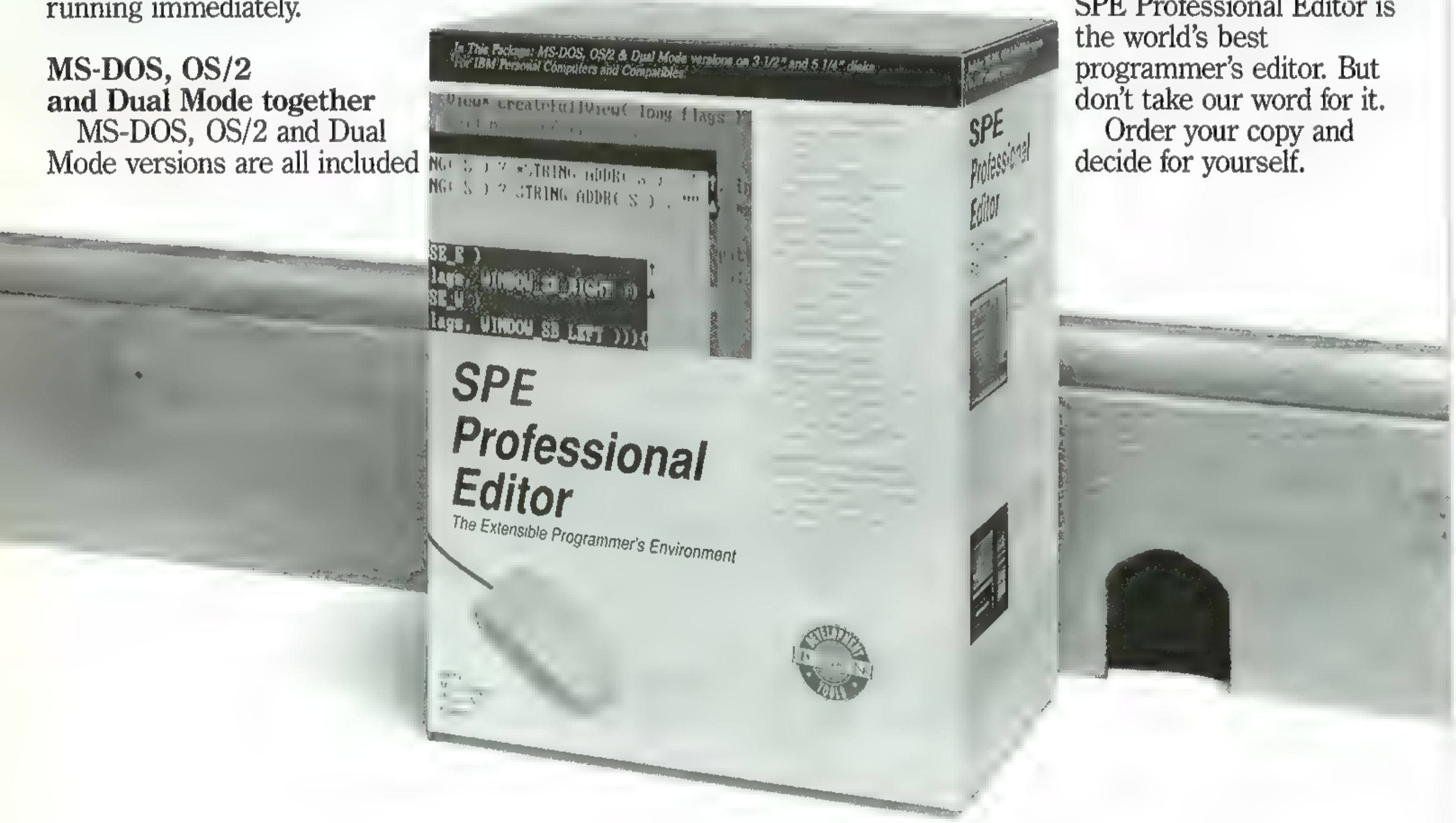
With or without a mouse

Use the editor with or without a mouse — all functions are available without lifting your fingers from the keyboard. But the click-select windows, scroll bars, zoom, shrink-to-icon, block text selects, and other speedy mouse actions make editing extremely efficient.

Don't take our word for it

We're confident that the SPE Professional Editor is the world's best programmer's editor. But don't take our word for it.

Order your copy and decide for yourself.


In This Package: MS-DOS, OS/2 & Dual Mode versions on 3 1/2" and 5 1/4" disks
For IBM Personal Computers and Compatibles
SPE
Professional
Editor
The Extensible Programmer's Environment

Works with or without a mouse.
Packaged with or without Microsoft® Mouse.

Version 1.1 now available

APPROVED DEALERS

Absolute Computing Limited, Absolute House
124 Leavesden Road, Watford, Herts. WD2 5EG. Tel: (0923) 242570

Computer Solutions Limited, Canada Road, Byfleet
Surrey KT14 7HQ. Tel: (09323) 52744

Grey Matter Limited, Prigg Meadow, Ashburton
Devon TQ13 7DF. Tel: (0364) 53499

Roundhill Computer Systems Limited, Orchard House
Ogbourne St. George, Marlborough, Wiltshire SN8 1SU. Tel: (0672) 84535

Reflex Technology Limited, 9 Buckingham Place
Bellfield Road, High Wycombe, Bucks HP13 5HW. Tel (0494) 465907

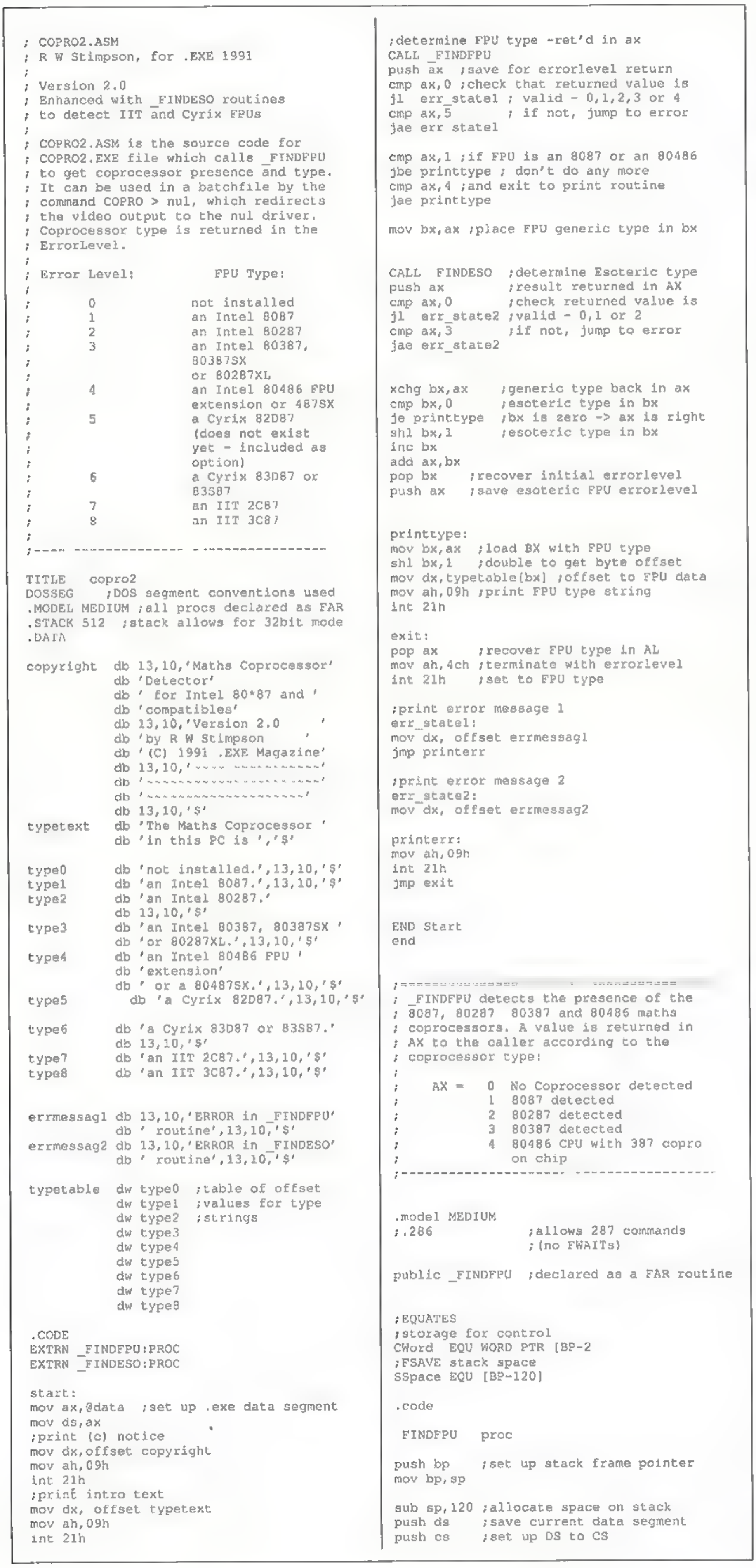
Frontline Distribution Limited, Intec 1, Wade Road
Basingstoke, Hampshire RG24 0NE. Tel: (0256) 463344

System Science, 3-5 Cynthia Street
London N1 9JF. Tel: 071-833 1022

Software Express, Portmill House, Portmill Lane
Hitchin, Herts SG5 1DJ. Tel: (0462) 422525

Software Generation · Portmill House, Portmill Lane, Hitchin, Herts SG5 1DJ

CIRCLE NO. 878



```
; COPRO2.ASM
; R W Stimpson, for .EXE 1991
;
; Version 2.0
; Enhanced with _FINDESO routines
; to detect IIT and Cyrix FPUs
;
; COPRO2.ASM is the source code for
; COPRO2.EXE file which calls _FINDFPU
; to get coprocessor presence and type.
; It can be used in a batchfile by the
; command COPRO > nul, which redirects
; the video output to the nul driver.
; Coprocessor type is returned in the
; ErrorLevel.
;
; Error Level:          FPU Type:
;
;       0             not installed
;       1             an Intel 8087
;       2             an Intel 80287
;       3             an Intel 80387,
;                     80387SX
;                     or 80287XL
;       4             an Intel 80486 FPU
;                     extension or 487SX
;       5             a Cyrix 82D87
;                     (does not exist
;                     yet - included as
;                     option)
;       6             a Cyrix 83D87 or
;                     83S87
;       7             an IIT 2C87
;       8             an IIT 3C87
;
;---- -------------- -----------------

TITLE   copro2
DOSSEG    ;DOS segment conventions used
.MODEL MEDIUM ;all procs declared as FAR
.STACK 512  ;stack allows for 32bit mode
.DATA

copyright  db 13,10,'Maths Coprocessor'
           db 'Detector'
           db ' for Intel 80*87 and '
           db 'compatibles'
           db 13,10,'Version 2.0      '
           db 'by R W Stimpson     '
           db '(C) 1991 .EXE Magazine'
           db 13,10,'~~~~ ~~~~~~~~~~~'
           db '~~~~~~~~~~~~~~~~~~~~~~'
           db '~~~~~~~~~~~~~~~~~~~~~'
           db 13,10,'$'
typetext   db 'The Maths Coprocessor '
           db 'in this PC is ','$'

type0      db 'not installed.',13,10,'$'
type1      db 'an Intel 8087.',13,10,'$'
type2      db 'an Intel 80287.'
           db 13,10,'$'
type3      db 'an Intel 80387, 80387SX '
           db 'or 80287XL.',13,10,'$'
type4      db 'an Intel 80486 FPU '
           db 'extension'
           db ' or a 80487SX.',13,10,'$'
type5        db 'a Cyrix 82D87.',13,10,'$'

type6      db 'a Cyrix 83D87 or 83S87.'
           db 13,10,'$'
type7      db 'an IIT 2C87.',13,10,'$'
type8      db 'an IIT 3C87.',13,10,'$'


errmessag1 db 13,10,'ERROR in _FINDFPU'
           db ' routine',13,10,'$'
errmessag2 db 13,10,'ERROR in _FINDESO'
           db ' routine',13,10,'$'

typetable  dw type0  ;table of offset
           dw type1  ;values for type
           dw type2  ;strings
           dw type3
           dw type4
           dw type5
           dw type6
           dw type7
           dw type8

.CODE
EXTRN _FINDFPU:PROC
EXTRN _FINDESO:PROC

start:
mov ax,@data  ;set up .exe data segment
mov ds,ax
;print (c) notice
mov dx,offset copyright
mov ah,09h
int 21h
;print intro text
mov dx, offset typetext
mov ah,09h
int 21h
```

```
;determine FPU type -ret'd in ax
CALL _FINDFPU
push ax  ;save for errorlevel return
cmp ax,0 ;check that returned value is
jl  err_state1 ; valid - 0,1,2,3 or 4
cmp ax,5       ; if not, jump to error
jae err state1

cmp ax,1 ;if FPU is an 8087 or an 80486
jbe printtype ; don't do any more
cmp ax,4 ;and exit to print routine
jae printtype

mov bx,ax ;place FPU generic type in bx


CALL  FINDESO  ;determine Esoteric type
push ax        ;result returned in AX
cmp ax,0       ;check returned value is
jl  err_state2 ;valid - 0,1 or 2
cmp ax,3       ;if not, jump to error
jae err_state2


xchg bx,ax    ;generic type back in ax
cmp bx,0      ;esoteric type in bx
je printtype  ;bx is zero -> ax is right
shl bx,1      ;esoteric type in bx
inc bx
add ax,bx
pop bx     ;recover initial errorlevel
push ax    ;save esoteric FPU errorlevel


printtype:
mov bx,ax  ;load BX with FPU type
shl bx,1   ;double to get byte offset
mov dx,typetable[bx] ;offset to FPU data
mov ah,09h ;print FPU type string
int 21h

exit:
pop ax     ;recover FPU type in AL
mov ah,4ch ;terminate with errorlevel
int 21h    ;set to FPU type

;print error message 1
err_state1:
mov dx, offset errmessag1
jmp printerr

;print error message 2
err_state2:
mov dx, offset errmessag2

printerr:
mov ah,09h
int 21h
jmp exit


END Start
end


;================       ============
; _FINDFPU detects the presence of the
; 8087, 80287  80387 and 80486 maths
; coprocessors. A value is returned in
; AX to the caller according to the
; coprocessor type:
;
;    AX =   0  No Coprocessor detected
;           1  8087 detected
;           2  80287 detected
;           3  80387 detected
;           4  80486 CPU with 387 copro
;              on chip
;---------------------  -----------------


.model MEDIUM
;.286              ;allows 287 commands
                   ;(no FWAITs)

public _FINDFPU  ;declared as a FAR routine


;EQUATES
;storage for control
CWord  EQU WORD PTR [BP-2
;FSAVE stack space
SSpace EQU [BP-120]

.code

 FINDFPU   proc

push bp    ;set up stack frame pointer
mov bp,sp

sub sp,120 ;allocate space on stack
push ds    ;save current data segment
push cs    ;set up DS to CS
```

Figure 3 - Detecting FPU types

They are 50% faster than the 287, because they divide the FPU clock by two instead of the older 287's factor of three. This means that 387 code can now be run on 286, 386, 386SX and 486 platforms.

Memory-mapped

All Weitek and the latest Cyrix EMC87 FPUs are memory-mapped. To the CPU, they appear as a valid segment of memory address space from 0C0000000h to 0C3FFFFFFh. Weitek occupies the full 64 KB, whereas Cyrix only uses the lower 4 KB to 0C0000FFFh. Systems without an 80386 can be used for memory-mapped FPUs but require a Virtual Memory Manager so that the address space can be mapped into HIMEM - the first 64 KB above the 1 MB 'barrier'. For 80386 virtual 8086s, the Page addressing must be enabled to access the Weitek segment. (The Microsoft EMS386.SYS driver has a switch to do just this.)

Processor read instructions from this memory are interpreted as data transfer commands, and processor write instructions as operation commands. This has the benefit over the Intel convention in being able to move data directly from the CPU to the FPU.

Weitek offer thirty-two 32-bit registers which can be combined to provide sixteen 64-bit double precision registers. The Weitek scores over the Intel series in taking fewer clock cycles to complete operations. It can be installed into systems which provide a superset of the Intel FPU socket, and can operate co-resident with an Intel FPU if the system board has both sockets provided. Other options include it on an add-in board.

The Cyrix EMC87 provides an identical register structure to the Intel standard and has the ability to accept both peripheral and memory-mapped commands. The major difference is that the Cyrix introduces its own memory-mapped protocol, so Weitek FPU code will not function. Until compiler designers offer wide and integrated support for this Cyrix FPU, it will remain a curiosity for the majority of application designers.

The Weitek FPUs also require compilers which offer specific support libraries, since Intel 80x87 code is ignored. These are widely available for C, FORTRAN and Pascal. Non-mainstream language support libraries can be written if required in assembler using assembly-level macros which Weitek can provide.

Detecting FPUs

The simplest, but least reliable, method is an MS-DOS INT 11 call for a list of installed equipment (See Figure 2a). The second detection method, Figure 2b, is far more robust, in that it attempts to initialise an FPU and checks the response to determine if it is operational. The FPU control word after FINIT is examined - it's 03FFh on an 8087 and 037Fh on all other FPUs. So if the upper byte does not equal 03h then no FPU is present. Reading the control word or tag word after FINIT and detecting its status can also be used.

The code shown in Figure 3 is a MASM V5.0 assembly routine which can be called by high-level languages to detect and distinguish between the 8087, 80287, 80387 and the 486 FPU extension. It also identifies the Intel compatibles discussed later. The following discussion describes how the code operates. The widely published 8087 and 80287 detection routines are included here for completeness.

The 8087 is the only FPU to respond to the interrupt bit in the FPU control word. By loading the control word with the bit cleared and then issuing the FDISI command (Disable FPU sourced Interrupts) an 80287 or 80387 is revealed if the bit does not change state.

The 80387 is detected by its behaviour when given the values -1/0 and 1/0 to calculate. The 80287 (and 8087) treat both these infinity values as the same (Projective infinity). The 80387 treats the two values as different (Affine infinity). If 1/0 returns a different response to that of -1/0 then the FPU is an 80387. The new 80287XL and 80287XLT chips ignore the infinity bit, hence they, (and the 387SX) are indistinguishable from the 80387DX.

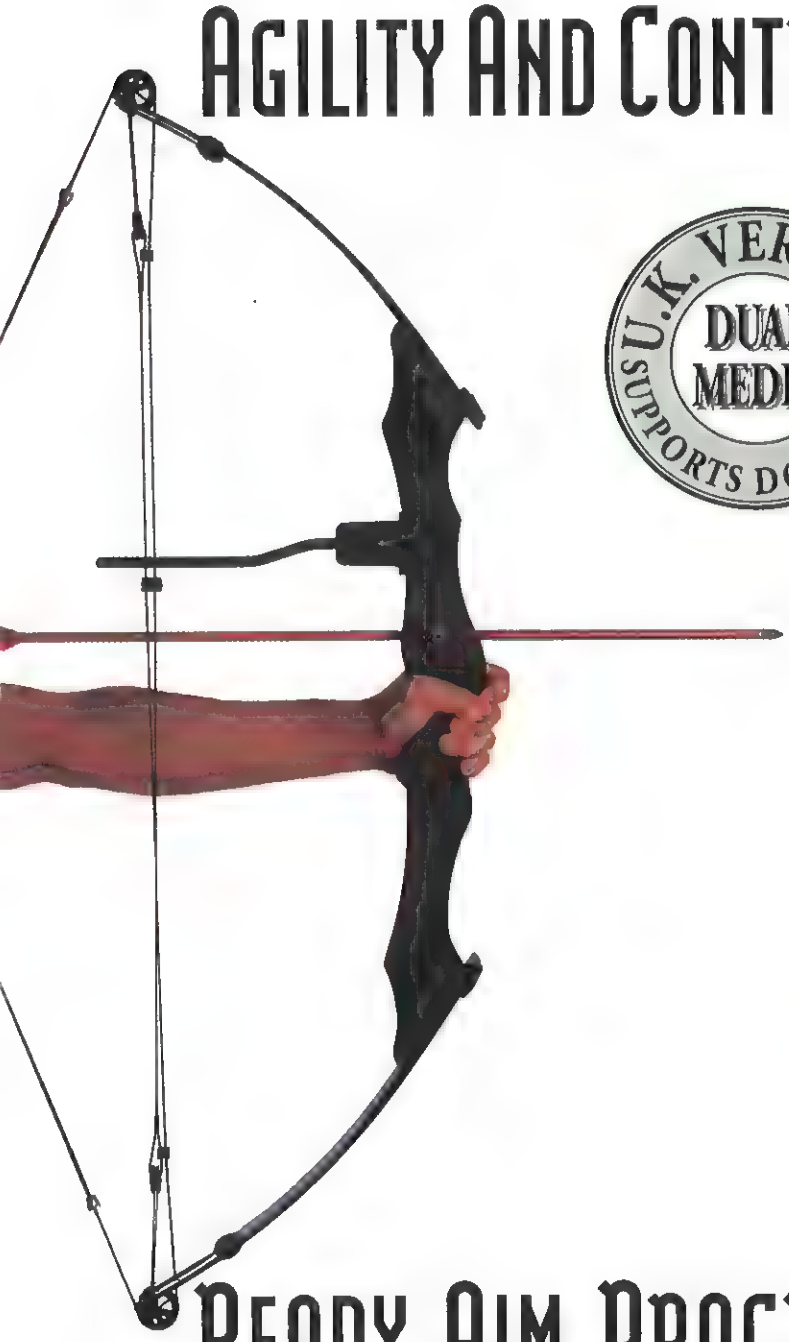
```
pop ds

cld         ;process upwards in memory
xor dx,dx  ;clear dx to 0
mov CWord,dx  ;clear controlword

;VERIFY FPU PRESENCE:
FWAIT       ;insert own waits for 8087
FNINIT      ;initialise coprocessor
FWAIT       ;
FNSTCW CWord  ;store control word
cmp BYTE PTR [BP-1],03h ;byte 03h?
jne fc_exit   ;if not, no copro
              ;CHECK FOR 8087:
mov dx,1      ;set FPU type in DX as 87
and CWord,0FF7Fh ;set interrupts
FWAIT
FLDCW CWord   ;load control word
FWAIT
FDISI         ;disable interrupts
FWAIT
FSTCW CWord   ;store control word
test CWord,0080h ;has bit changed?
jnz fc_exit   ;if set to 1 its an 8087

;CHECK FOR 80287:
mov dx,2     ;set FPU type in DX as 287
FINIT         ;initialise copro
FLD1          ;load 1
FLDZ          ;load 0
FDIV          ;divide 1 by 0
FLD  ST       ;push copy of infinity
              ;on FPU stack
FCHS          ;change sign of top copy
FCOMPP       ;infinity values compared
              ;in bits C0,C1,C3 of SW
FSTSW CWord   ;MODIFIED FROM FSTSW
              ;AX and 286 removed
mov ax,CWord  ;load status word to AX
              ; (287/387)
sahf          ;store flags
              ; C3 sets zero flag
jz fc_exit    ;if zero, its a 287

;CHECK FOR 80387:
mov dx,3     ;set FPU type in DX as 387

CALL CHECKMODE  ;CPU op mode using
                ;FSAVE data
CALL FIND486    ;distinguish 387 and 486

fc_exit:
pop ds          ;recover original value
mov ax,dx       ;return FPU type in AX

mov sp,bp       ;restore stack
pop bp
ret

 FINDFPU       endp

; --------------------------------------
; FSAVE ADDRESS DATA BYTES
;---------------------------------------

for16bitr   db 7,6,7,89h
            db 0aH,0bH,0ch,8dH
for16bitp   db 8,6,7,8
            db 9,0aH,0bH,0cH,0dH
for32bitr   db 0aH,0cH,0dH,91h,12h,13h
            db 14h,15h,19h,1aH,1bH
for32bitp   db 0dH,0cH,0dH,0eH
            db 0fH,10h,11h,93h,14h,15h
            db 16h,17h,18h,19h

 ChkTable     dw for16bitr
              dw for16bitp
              dw for32bitr
              dw for32bitp
;--------------------------------------
```

```
; CHECKMODE examines the FSAVE data
; structure to determine whether CPU is
; in 16 bit real, 16 bit protected,
; 32 bit real or 32 bit protected mode.
;
; BX is set with a value indicating the
; CPU mode:
;
; 0 = 16 bit real; 1 = 16 bit protected
; 2 = 32 bit real; 3 = 32 bit protected
;
;--------------------------------------
CHECKMODE      PROC NEAR

FINIT             ;initialise copro again
FINCSTP            ;status word non-zero
FINCSTP            ;by making stk ptr = 3
FINCSTP            ;so h byte of SW = 18H
FSAVE SSpace       ;copro state to stack

;TEST FOR 16/32 BIT MODE:
;status word hibyte=18h?
cmp BYTE PTR SSpace+3,18h
je is16bit386 ;yes, 386 in 16 bit mode

mov bh,1       ;no, 386 in 32 bit mode
jmp testrealmode

is16bit386:
mov bh,0      ;set value for 16 bit mode

;TEST FOR REAL/PROTECTED MODE:
testrealmode:
cmp bh,0
je test16bitreal
;we're testing 32 bit structure
;load opcode word area
mov ax,word ptr SSpace+10h
and ax,07ffh  ;any bits are set
cmp ax,1F7h   ;is it correct opcode?
je  isrealmode ;in real mode
jmp isprotmode ;or protected mode

test16bitreal:
;load opcode word area
mov ax,word ptr SSpace+8h
and ax,7FFh   ;check for set bits
cmp ax,1f7h   ;is it the correct opcode?
je isrealmode ;if equal, then real mode

isprotmode:
mov bl,1       ;it is protected mode
jmp setupbx

isrealmode:
mov bl,0       ;it is real mode

setupbx:
shl bh,1      ;make BX between
add bl,bh     ;0 and 3
xor bh,bh

RETN
CHECKMODE      ENDP

;--------------------------------------

; FIND486 contains the code to
; distinguish the 386 series from the
; 486 series coprocessor by check-
; ing the instruction pointer and CS
; selector bits for zero contents after
; a second FSAVE. (First was in
; CHECKMODE).
; Routine is called with BX is set to a
; value indicating the CPU mode:
;
; 0 = 16 bit real; 1 = 16 bit protected;
; 2 = 32 bit real; 3 = 32 bit protected
;--------------------------------------
```

```
FIND486        PROC NEAR

push ds             ;save data segment
push cs             ;load code segment
                    ;into data
pop ds
                    ;get FPU into a known
                    ;state:
finit               ;initialise copro
fldpi               ;load PI 3.14159...
fldlg2              ;load log(10) of 2
fadd                ;add together


fsave SSPace        ;save FPU status
fsave SSpace        ;save data area again
; address bytes should now be zero
; after two FSAVEs
fwait


;load table for mode in BX
mov ax, word ptr CS:ChkTable[bx]
mov bx,ax    ;into BX for XLAT
xor ax,ax    ;AL = 0 for first byte
xlat         ;load table length
mov cx,ax    ;load into CX as counter
xor ax,ax    ;reset ax to zero

testfor486:
inc al       ;look at next test byte
push ax      ;save byte location pointer
push bx      ;save table pointer
xlat         ;load byte into AL
test al,80h ;is bit 7 set?
jz stdtest  ;if not, std check

and AL,7Fh  ;special check
cmp al,13h  ;is it byte 13h?
jb bits4to7 ;if not, it must be
            ;9,D or 11
mov ah,0F8h ;it is byte 13h
jmp test_bit;test the bits

bits4to7:
mov ah,0F0h ;set appropriate bits

test_bit:
mov bl,al  ;load BX with pointer to byte
xor bh,bh  ;and save in si
mov si,bx
;load byte from FSAVE table
and ah, byte ptr SSpace[si]
jmp chk4zero  ;check byte is clear

stdtest:
mov bl,al   ;byte from FSave table
mov si,bx   ;point SI to the offset
mov ah,byte ptr SSpace[si] ;and transfer

chk4zero:
cmp ah,0    ;is byte in FSAVE area zero?
pop bx      ;recover table pointer
pop ax      ;recover location pointer
jne exit386 ;byte is not zero, so exit
loopz testfor486

mov dx,4    ;it must be a 486

exit386:
pop ds
retn

FIND486        ENDP
               end
```

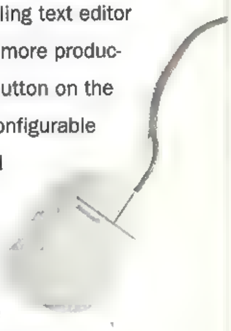
Figure 3 - Detecting FPU types (continued)

BRIEF'S NEW MOUSE SUPPORT GIVES YOU GREATER AGILITY AND CONTROL.



U.K. VERSION
DUAL MEDIA
SUPPORTS DOS & OS/2

More than 100,000 programmers around the world swear by the virtues of BRIEF – its unmatched power, flexibility, and intuitive command structure. Now, thanks to mouse support and other new features, version 3.1 of the best-selling text editor will make you quicker and more productive than ever. ¶ Hit a button on the mouse – a special, user-configurable menu window pops up and lets you execute any command you wish. Mark text and move around in your windows without using the keyboard. Zoom your windows to fill the screen. ¶ As snappy as it is, the mouse is only one enhancement among many. BRIEF 3.1 also takes advantage of EMS memory whenever it reads in a file. The result: faster performance, whether you're editing a file or running a macro. ¶ So what else is new? Redo – a step for step complement to undo. Support for Microsoft C 6.0 Advisor and Microsoft's Programmer's Workbench. And more. ¶ BRIEF's seamless interface with other Solution Systems products gives you instant access to the tools you need. A single keystroke lets you keep track of product development with Sourcerer's Apprentice – Version Control for the Professional. Analyze product performance with CHARGE. And nothing beats dBRIEF for tailoring BRIEF to Dbase or Paradox. ¶ *BRIEF sells for £199 excluding VAT.* ¶ *Once you start working with BRIEF 3.1 you'll really fly.*



READY, AIM, PROGRAM.

Order your copy of BRIEF today.
call:
0763 244141

Solutionsystems
THE PHYSICS OF PROGRAMMING
1 The Maltings, Green Drift,
Royston, Hertfordshire SG8 5DB
Telephone: 0763-244141
Facsimile: 0763-244025

BRIEF requires an IBM PC, AT, or 100% compatible, 256K RAM, and two disk drives.

CIRCLE NO. 879

Although the FPU extension within the 486 is an 80387, detection means that you can identify the 486 and take advantage of the additional CPU functions. The method presented here is more complex than the others since it is designed to work in all 486 modes - Real, Protected, Virtual 86, 16- and 32-bit.

Existing published methods suggest changing the newly defined bits in EFLAGS of the 486. This is privilege level sensitive Virtual 86 mode, so it cannot be universally applied without restrictions. Surprisingly, Intel demonstrates this method in the latest 487SX specification - but there is a more rigorous and universally applicable technique which is independent of the need to know processor type - and uses Intel-published information *(Intel 486SX Microprocessor/487SX Math Coprocessor Specification, April 1991, Part Number 240950-001).*

The i486 and 487SX clear the instruction and data pointers in the FPU after an FSAVE command. So if FPU operations are made and the state FSAVEd then immediately FSAVEd again, the pointers should be zeroed in the second state. If these zeros are detected then the 80486 FPU or 487SX is present.

The FSAVE data structures differ between the CPU modes. The detection method used creates an FSAVE signature, determines CPU mode, and finally checks for zeroed pointers. Some of the bytes to be checked contain partial bit-fields. These are flagged in the routine by setting bit seven in the data byte defining the position in the FSAVE data structure - the routine detects then strips this off and applies an AND with the correct byte to verify the FSAVE bits are zero.

386 Mode

The 16/32-bit mode test checks if the Status Word is the second in the FSAVE structure by looking for an 18h signature in the high byte set up with three FINSTP instructions. The test for real or protected mode is made by looking at the 10-bit opcode area which is located in bytes 8 and 9 in 16-bit mode and bytes 10 and 11 in 32-bit mode. The FPU stores the lower of the opcode, which here is FINSTP - D9F7h (stored as 1F7h). This method avoids loading the MSW, which is not done here to avoid any possible privilege level exception error in Protected or Virtual 8086 modes.

Compatibles

AMD, Cyrix and Integrated Information Technology (IIT) all produce Intel-compatible alternatives. AMD currently offers the 80C287 - a 287 clone which is simply around half the price of the comparable Intel chip. AMD literature does not provide any performance comparisons, but if price is the only guide, it is possibly one of the best price-performance boosts a 286 PC could receive.

```
;=================================================
; ** DO NOT RUN THESE CALLS UNLESS **

; ** _FINDFPU RETURNS 2 OR GREATER **

; _FINDESO detects the presence of IIT
; and Cyrix FPUs. A value is returned in
; AX to the caller according to the FPU
; type:
;    AX =   0  Intel, 1  Cyrix, 2  IIT

; The following routines are true (AX=1)
; if the FPU is present.
; extern boolean _IITCoPro(void)
; extern boolean _CyrixFPU(void)
; extern boolean _CyrixEMC87(void)
;-------------------------------------------------

.model MEDIUM    ;MEDIUM - BASIC, LARGE
;all others
.286


;EQUATES


;storage for control
CWord  EQU WORD PTR [BP-2]
;FSAVE stack space
SSpace EQU [BP-120]
;ENV save space
EnvSpace EQU [bp-28]


;stack data area
Stk10rT EQU TBYTE PTR [bp-12]
;as two types
Stk10rW EQU WORD PTR [bp-4]

.code
PUBLIC  _FINDESO, _CyrixFPU
PUBLIC  _IITCoPro, _F4X4
PUBLIC  _CyrixEMC87

mem04i3 dd   3 ;two byte integer

mem10r1 dt   1 ;ten byte real (inf)


;=============================================
; Following Code is Provided by IIT on
; Disc with all IIT Coprocessors;

_IITCoPro PROC FAR

FINIT              ;initialise copro

;load 10 byte real 1 (invalid)
FLD   cs:mem10r1
;add top of stack to itself

FADD    ST(0),ST
FSTSW   ax  ;status word to ax (287/387)
FWAIT       ;let FPU sync with CPU

and     ax,02 ;Intel has bit 1 set sts w
mov     ax,0  ;prepare to return 0

jnz     NotIITexit ;if not set, not IIT

mov     ax,1  ;Indicate IIT CoPro


NotIITexit:
retf            ;C return convention


_IITCoPro ENDP


;=============================================


_CyrixFPU PROC FAR

push bp       ;set up stack frame pointer
mov bp,sp
sub sp,12     ;allocate space on stack
push bx       ;clean and tidy start


FINIT           ;start with clear FPU
FLDZ            ;load ST(0) with 0
FSTP Stk10rT    ;FPU to known state
FILD cs:mem04i3;3 into ST(0)
F2XM1           ;calc 2^3 - 1 (creates
                ;an out-of-range error)
FSTP Stk10rT    ;save result to stack
FWAIT           ;time for FPU
mov ax, Stk10rW  ;load mantissa
mov bx,0        ;prepare to return 0
cmp ax, 3FFFh  ;is mantissa zero?
je  found_Eso  ;if yes, not Intel
cmp ax, 3FFEh  ;or is mantissa -1?
jne NoCyrixExit;not Cyrix


found_Eso:      ;distinguish Cyrix
xor ax,ax
mov al, byte ptr [bp-5]
cmp al,0C0h     ;value unchanged?
jne NoCyrixExit
mov bx,1        ;found Cyrix FPU

NoCyrixExit:
mov ax,bx       ;recover code from bx
pop bx          ;restore bx


mov sp,bp       ;recover local data
pop bp
retf            ;C return convention


_CyrixFPU ENDP


 CyrixEMC87 PROC


CWord equ word ptr [sp-2]


push bp
mov bp,sp
sub sp,2        ;set up stack space
push bx


fstcw CWord   ;save control word
mov bx, CWord ;copy to bx
or byte ptr [sp-1] ,80h ; set high bit
fldcw CWord   ;reload modified word
fstcw CWord   ;save word from FPU
xor ax,ax     ;clear ax
mov al, byte ptr [sp-1] ;load high byte
and al,80h    ;check if high bit set
cmp al,0
je EMC87exit  ;AL is zero -> not found


mov CWord,bx  ;restore original
fldcw CWord   ;control word
mov ax,1      ;set to show found EMC87


EMC87Exit:
pop bx
mov sp,bp
pop bp
retf


_CyrixEMC87 ENDP


;=============================================

_FINDESO PROC FAR

CALL _CyrixFPU ;Cyrix installed?
cmp ax,1       ;yes AX=1, no AX = 0

jne TryIIT     ;if not,try IIT test

jmp EsoExit

TryIIT:
CALL _IITCoPro ;try IIT test
cmp ax,1       ;is it IIT?
jne EsoExit    ;if not,exit
mov ax,2       ;it's IIT
;exit AX = 0 (no FPU) AX = 2 (IIT)


EsoExit:
ret
_FINDESO ENDP
```

Figure 4 - Detecting Esoteric FPU Types

Borland C++ wants glockenspiel CommonView® 2

You've bought Borland C++, eager to build your first Windows application. However, writing "Hello World" isn't so easy, is it?

Even with Borland C++ you still need to learn quite a lot about Windows programming.

But hang on a minute!

Why re-invent the Window?

Why waste time learning how Windows works when you could be concentrating on the application itself.

What you need is Glockenspiel CommonView 2 – the C++ class library that models the real world of Windows.

CommonView encapsulates most of Windows in about 70 classes. You don't have to worry about the internals of Windows – CommonView takes care of that. That means a substantial code reduction, not to mention easier, more robust programming.

CommonView goes further.

Using Borland C++, you extend the CommonView classes to meet your own requirements. A few extra lines of C++ code and you could, for example, turn an edit control into a password control. Then reuse it in other applications.

And CommonView applications are portable to other GUIs. (Other CommonView versions are currently available for PM and OSF/Motif.)

Glockenspiel CommonView is regarded as the standard in the world of Windows development (using Glockenspiel C++). Thousands of developers use it to speed up and simplify their projects – did you really think they'd stick with the API?

Now it's your turn!

The Efficient Choice

Glockenspiel CommonView won the Computer Language Productivity Award of 1990. Here's why:

■ **Develop Quickly**

Glockenspiel CommonView works with Borland's Programmer's Platform. Write, browse and debug your source code efficiently and quickly.

■ **Produce Immediately**

The CommonView 2 class library lets you begin developing straight away. Use C++ to extend the existing class hierarchy to your own requirements.

■ **Build with confidence**

CommonView 2 is an extensible C++ library. It exploits the strong type-checking features of C++ to produce robust, bullet-proofed classes that you extend in confidence.

■ **Keep it small**

CommonView 2 applications share the compact CommonView DLL. So you're already starting off with a keen edge on efficiency. C++ keeps the application size to a minimum – apps begin at a mere 5k!

■ **Watch it move!**

CommonView apps incur little overhead. They're fast, efficient and smooth-running.

Send for FREE Illustrator

CommonView 2 is simple to use and easy to learn. Now, with the free CommonView Illustrator and hypertext help, you can quickly learn how windowing features are implemented and manipulated in CommonView 2 using Borland C++. **How to get it:** Fax **QA Training** or download it from the Byte DemoLink on BIX (617-861-9767).

Specifications:

CommonView 2/HT for Borland C++ requires Borland C++ 2.0

CommonView 2/HT consists of over 70 classes Includes Help classes which enable applications to use Winhelp files.

Extensive documentation includes CommonView tutorial and class library reference manual.

CommonView 2/HT costs £399. Ships with the CommonView Illustrator.

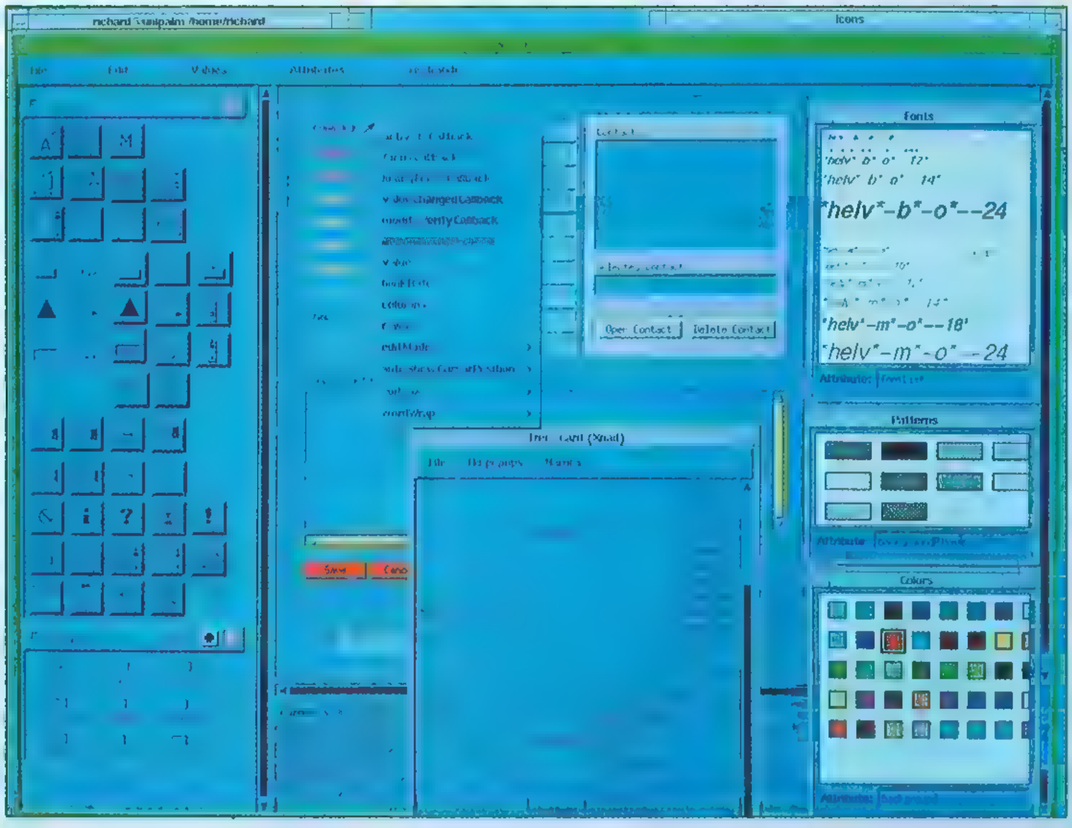
glockenspiel
class constructors

To order:

QA TRAINING,
Cecily Hill Castle, Cecily Hill, Cirencester, Glos GL7 3EF, UK
Telephone: +44 (0)285 655888. Fax: +44 (0)285 640181

Glockenspiel CommonView and Glockenspiel C++ are registered trademarks of Glockenspiel Ltd. The trademarks of their respective corporations are acknowledged.

CIRCLE NO. 880



See it on IBM's stand Software Tools '91

TeleUSE

The OSF/Motif user interface builder

"With TeleUSE, the work to develop user interfaces can be reduced by up to 90% – design quality is improved and the development work becomes more enjoyable."

Unipalm are also at NETWORKS '91 NEC 4-6 June ***Stand 298***

&

The UNIX SHOW Olympia 18-20 June ***Stand 111D***

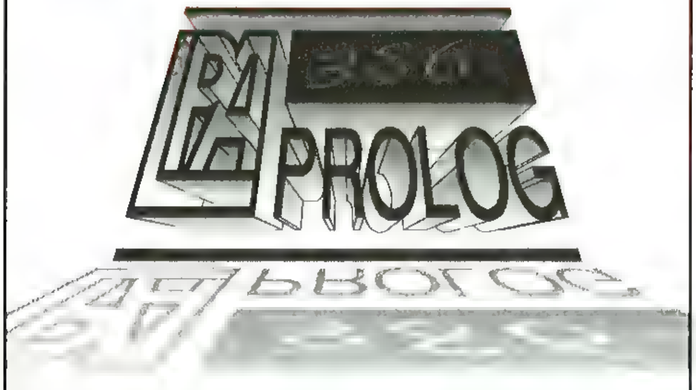
Unipalm

Unipalm Limited 145-147 St Neots Road Hardwick Cambridge CB3 7QJ England
Tel 0954 211797 Fax 0954 211244 Intl Tel +44 954 211797 Intl Fax +44 954 211244

TeleUSE registered trademark TeleSoft AB Sweden. OSF, OSF/Motif, Motif trademarks Open Software Foundation, Inc. IBM registered trademark International Business Machines Corporation.

CIRCLE NO. 881

Think BIG!



Forget 640K. Forget EMS. Forget the 16-bit squeeze. With LPA 386-PROLOG, you get a genuine 32-bit programming environment, which runs directly under MS-DOS and yet is capable of addressing up to 4G (4096M) of memory. All you need is a 386 or 486, and the talent to Think BIG!

LPA

Logic Programming Associates Ltd
Studio 4, Royal Victoria Patriotic Building
Trinity Road, London, SW18 3SX, England
Tel: 081 871 2016 - Fax: 081 874 0449

CIRCLE NO. 882

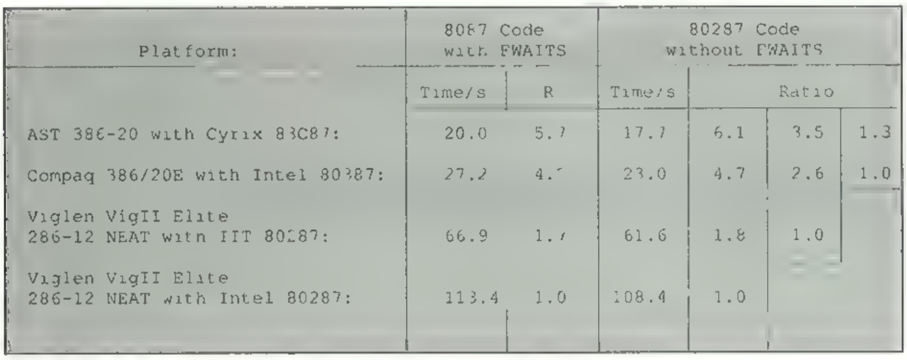
8051 80X86 68XXXX In-Circuit Emulation

- ❑ Non-Intrusive Emulation
- ❑ Performance Analysis
- ❑ High Level Language Debugging
- ❑ Leasing
- ❑ Rental
- ❑ Rent-Purchase

For Full Details and Interactive Demo Disk, please contact:
Hitex (UK) Ltd, Science Park, Coventry CV4 7EZ
☎ Tel: (0203) 692066 Fax: (0203) 692131

hitex
Experience in High Technology

CIRCLE NO. 883



Platform:	8087 Code with FWAITS		80287 Code without FWAITS			
	Time/s	R	Time/s	Ratio		
AST 386-20 with Cyrix 83C87:	20.0	5.7	17.7	6.1	3.5	1.3
Compaq 386/20E with Intel 80387:	27.2	4.2	23.0	4.7	2.6	1.0
Viglen VigII Elite 286-12 NEAT with IIT 80287:	66.9	1.7	61.6	1.8	1.0	
Viglen VigII Elite 286-12 NEAT with Intel 80287:	113.4	1.0	108.4	1.0		

Figure 7 - FPU Comparison Table

Cyrix manufactures i387 and 387SX clones (the CX-83D87 and CX-83S87) which process values with better accuracy (one or two LSBs closer to the true result in extremes) and lower clock-cycle counts than the Intel equivalents. The Cyrix FPUs use algorithms implemented in on-chip hardware rather than the microcode used by Intel. Additionally, the error states the FPU can enter are more precisely defined, resulting in known, testable error conditions.

Although uniquely identifiable as Cyrix using software, the benefits are reaped by the user rather than the program developer. This will not affect code design, but could affect your recommendations for preferred hardware and presenting benchmarking values.

A simple routine is presented in Figure 4 to detect the Cyrix FPU. If the $y = 2^x - 1$ or F2XM1 instruction is executed, with x outside the range $-1.0 \leq x \leq +1.0$, Intel FPUs leave the exponent and mantissa in an undefined state. Cyrix, however, sets the exponent to 3FFEh or 3FFFh and leaves the mantissa untouched. (Incidentally, IIT sets the exponent in an identical manner but leaves the mantissa in a different state. This fact could be used to detect the Intel/Cyrix/IIT FPU origin, but I have chosen to stick with the IIT recommended detection technique.)

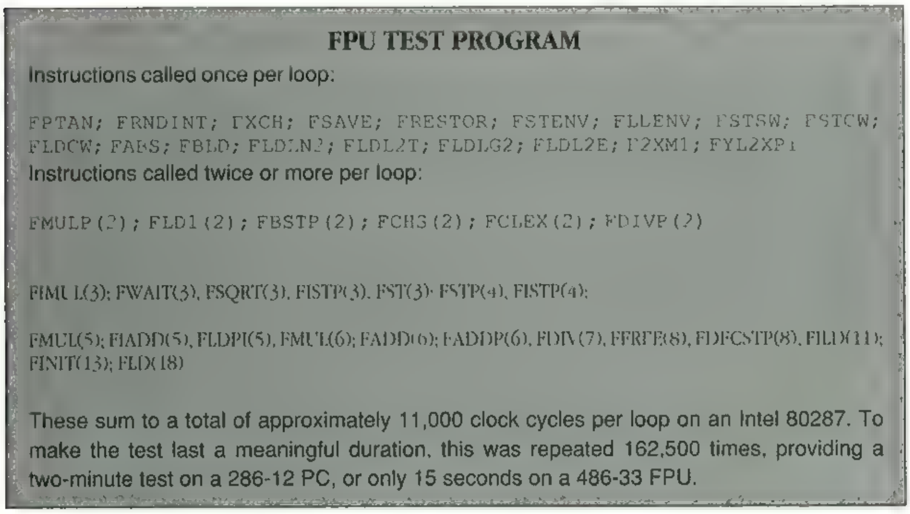
The Cyrix EMC87 can be detected by its ability to set the MSB of the FPU Control Word. This may cause long-term compatibility problems, because it is currently undefined by Intel.

IIT

IIT provides greatly enhanced 80287 and 80387 compatibles (the IIT-2C87 and IIT-3C87). The IIT-2C87 implements the full 80387 instruction set with similar speed benefits to the other Intel compatibles mentioned earlier. All IIT FPUs also provide three accessible banks of eight 80-bit registers, with a further internal set of eight 80-bit registers. This provides the option to have three calculation sets stored in the FPU's registers. The IIT FPU can simply switch banks of registers - saving 684 clock cycles (27 µs on a 25 MHz CPU) over the equivalent Intel FSAVE/FRSTOR action.

IIT FPUs recognise denormalised numbers as zeros. The consequent inability to set the denormalised exception flag in the FPU status word distinguishes the IIT, as shown in Figure 4.

IIT has extended 387 functionality further by using a fourth set of registers (which are inaccessible to the programmer) to permit vector multiplication by a 4 x 4 matrix (the size commonly required for 3D graphics operations) using only one FPU command, once the registers are loaded with matrix element values. The ability to process this multiplication-intensive problem with a single instruction in hardware reduces the calculation time by a factor of four or so when compared with a conventional 80387.



FPU TEST PROGRAM

Instructions called once per loop:

FPTAN; FRNDINT; FXCH; FSAVE; FRESTOR; FSTENV; FLLENV; FSTSW; FSTCW; FLDCW; FABS; FBLD; FLDLN2; FLDL2T; FLDLG2; FLDL2E; F2XM1; FYL2XP1

Instructions called twice or more per loop:

FMULP(2); FLD1(2); FBSTP(2); FCHS(2); FCLEX(2); FDIVP(2)

FIMUL(3); FWAIT(3), FSQRT(3), FISTP(3), FST(3), FSTP(4), FISTP(4);

FMUL(5); FIADD(5), FLDPI(5), FMUL(6); FADD(6); FADDP(6), FDIV(7), FFREE(8), FDECSTP(8), FILD(11); FINIT(13); FLD(18)

These sum to a total of approximately 11,000 clock cycles per loop on an Intel 80287. To make the test last a meaningful duration, this was repeated 162,500 times, providing a two-minute test on a 286-12 PC, or only 15 seconds on a 486-33 FPU.

Figure 8 - Test Suite FPU calls

Conclusions

To evaluate these FPUs with reasonable objectivity, and to provide a rudimentary guide to relative performance, I devised a test program which consisted of two hundred 287 instructions covering most of the instruction set in realistic problems (see Figure 8 for instruction set). This was then repeated in a loop and the elapsed time calculated. The test was run on AST, Compaq and Viglen hardware, using Intel, Cyrix and IIT FPUs. The IIT 2C87 returned performance times consistently 75% faster than the Intel 287. The test run on a Compaq 386/20E with an Intel 80387 and an AST 386-20 with a Cyrix 83D87 indicated a 30% improvement in performance between Cyrix and Intel - but this is not a perfect comparison.

If you are contemplating using the FPU in your program design, the choice is wider than ever before. I hope this has provided some insight into a little understood but widely available hardware upgrade which can transform software performance. My own personal conclusion is that best value is the cheapest FPU on offer - which is also likely to operate faster than the Industry standard Intel offers.

EXE

Bob Stimpson is a Project Manager concerned with real-time control of Industrial applications with a Systems House based in the North West. His personal interest lies in techniques which boost PC performance and provide useful information on PC configuration.

Many thanks to Alan Milosevic of DMST (0635 247100) for the IIT code examples and to Ambar Cascom (Cyrix 0296 434 141), Micro Call Solutions (Weitek 0844 261 500), DatronTech (AMD 0252 313 155) and Intel (0793 696000) for the extensive literature they kindly provided. Thanks also to Tim Frost of Roundhill Computer Systems for testing the code on his 486.

CUA - Who needs it?

Do we actually need the CUA to write Windows programs?
If not, why does Microsoft support it, and if so, why doesn't the company stick to it?

As I write my programs, I keep three books beside me. One is Petzold's book about programming Windows, one is the Kernigan and Ritchie book about C (some things one just *can't* avoid!), and the last is the CUA specification.

CUA stands for Common User Access. It is IBM's attempt to codify the standard look and feel which is now enshrined in Windows, OS/2 and other systems. It is not a big book, and on first appearance is not particularly deep, but it is a very well-crafted document, and it holds within it a number of very subtle assumptions.

In the old days of programming (it seems like only 20 years ago...) a program had to do a job - it required certain input to do that job, and it generated certain output as a result of it, but the dialogue between the user and the computer was given little, if any attention. Computers acquired the (justifiable) reputation of being difficult to use. Graphical environments promised to change all that, but at first failed abysmally. With half a dozen programs on the screen, all operating their own way, the user had all the old problems of remembering how to use his different pieces of software. On top of that, each program required an extra level of control to move it around on the screen. Pity the poor user who had to switch from one operating mode to another at the poke of a mouse!

Enter CUA. CUA defines a common grammar which all programs in the system use, so that this switching of operating modes is kept to a minimum. Certainly, CUA is flawed. Equally certainly, CUA is out of date. Even now, HCI researchers are still arguing what form this grammar should take. But even bad standards are better than no standards at all, and whatever else it is, CUA is a standard. Indeed, the Windows SDK provides a copy of the CUA document, and contains explicit support for most of the elements in it (albeit, sometimes in a rather roundabout manner), demonstrating that Microsoft wants programmers to follow the specification. At the same time, controlling Windows at any deeper level is profoundly awkward - if one tries to do anything remotely out of the ordinary, Windows sits in a corner and sulks - and the SDK has gone to some lengths to obscure much of Windows' real functionality. Crafting a new control can be a major undertaking, not least in navigating around the documentation.

Although Word for Windows' performance is nothing short of magnificent, its handling is the pits

I am writing this article with Word for Windows, a Microsoft product. Although its performance is nothing short of magnificent, its handling is the pits - its user interface bears no relation to the CUA that I have. It is full of funny modes, its controls (some of which work like 'radio buttons', and some like 'check boxes') look identical, and are bundled together in the same area of screen. They have pictures on them: some are well worth having, but most are silly and/or irrelevant and/or inconsistent. There are also status windows that don't tell you what status they are reporting. All this decoration takes up real estate on my screen which I could well use for other things. Even the help system is different from the standard Windows help, and is actually a poorer product for it, and the tutorial doesn't work (it crashes my machine repeatedly). Of all the programs I have ever used under Windows, Word is the only one whose manual I have had to read (and what a manual!). And why has Microsoft chosen to disregard the rules so flagrantly? (They are MS's own rules, after all.) Apparently so that the Word control panel looks like the flight deck of an aeroplane. So that of all the applications running on a Windows system, the one from Microsoft looks the prettiest.

Now I know it is terribly fashionable to complain about Microsoft, but this annoys me. I work very hard to get my programs working properly, and usable by my customers. I committed to Windows in the heady days of version 2 because it gave me the best way to achieve the functionality I wanted. Now Microsoft itself spoils the idea by apparently wanting to upstage anything I can do. But that is not the real issue here - I suspect that I am not alone in being annoyed by Microsoft, and one more annoyance isn't going to make much difference. No, the real issue is that Microsoft's handling of Word, its writing of manuals (which I have always found 'disappointing'), and even its investment in its support lines suggest to me that it really doesn't think about its end users. I feel sure that there is nobody at Microsoft (at least, nobody in any position to make decisions) who knows the first thing about user interfaces. Microsoft now has control of CUA, and is making inroads into other companies' GUIs, and that worries me.

EXE

Jules May tries to write Windows programs to deadlines which would be practical on any other operating system. He can be contacted on CIX as jules, or on 0707 44185 if he's not trying to get through to Microsoft technical support.

The views expressed in this article are Mr May's own, and do not necessarily represent the views etc.

Utilities and Programming Tools from Roundhill Computer Systems

.RTLink Plus

Pocket Soft's new "**Virtual Memory Linking**" provides the industry's most advanced memory management technology for Microsoft C, Microsoft Fortran and Clipper applications, with no changes to source code. **.RTLink Plus** also includes a profiler and supports multiple nested overlays, EMS or XMS caching and Codeview. It works with any language which produces standard .OBJ files. The normal price is £295.00 but until June 30th, 1991 Roundhill has a **special offer** price for .RTLink Plus of **£195.00** (plus VAT).

The Periscope Company, Inc.

Periscope/EM is a new product that offers a software-only solution for debugging programs which leave no room in the lower 640K. Qualitas and Periscope have worked together to allow Periscope to use protected extended memory when running in a 386 or 486 system with **386MAX** or **BlueMAX**. Periscope/EM costs only **£155.00** (plus VAT). For 386MAX/BlueMAX add £70.00/£90.00 (version 5.11 is required). Roundhill can also supply the full range of **Periscope hardware debuggers** - please call for details.

PANEL Plus II

Roundhill's **PANEL Plus II** screen manager includes an interactive screen design editor, C and MS-Fortran code generators, and a extensive library of user-interface functions. Support is included for pop-up fields and windows, multiple-line fields, scrolling regions, mouse and scroll-bar operations, pull-down menus and custom validation. There are interfaces to popular PC graphics libraries which allows operation in graphics mode under DOS and OS/2. Key assignments and edit functions are **CUA/SAA** compatible. PANEL Plus II comes with **full library source code**, and versions are available for DOS, OS/2 (including DLL), Unix, Xenix, QNX, and VMS. The library source and your application screens can be ported to any of these operating systems with **no royalties**. PANEL Plus II costs **£285.00** (plus VAT) per development licence. Please call for a free demonstration diskette.

Roundhill also stocks many other development tools: From **Copia International**, we have **AccSys for Paradox** and **AccSys for dBase IV** — easy-to-use, high performance interfaces between the database and your own C-language applications. From **Graphic Software Systems**, we have the **Graphics Development Toolkit** and **Graphical Kernel System** for either DOS or OS/2, as well as **XVT** for source-compatible GUI development over a wide range of operating environments. We also stock **Microsoft** language products. Roundhill is an authorised dealer for **Sage** (USA) and can supply the new **SPE editor** for DOS or OS/2 (including mouse support and unlimited undo/redo), and the **PVCS** source control software, which is now available for DOS, OS/2 several Unix platforms. Please call for a full catalogue and price list.

Roundhill Computer Systems Limited

Orchard House, Ogbourne St. George, Marlborough, Wiltshire SN8 1SU, England
Telephone: **0672 84 535** • Fax: 0672 84 525

SOFTWARE TOOLS

CIRCLE NO. 884

PostScript (P.S. it's a programming language too!)

PostScript is famous in the world of electronic printing - but how does it work as a programming language? Corinna Kinchin explains.

The PostScript programming language was developed by Adobe Systems Inc, a small company set up in 1982 by two ex-employees of Xerox PARC, Chuck Geschke and John Warnock. Geschke and Warnock intended PostScript primarily as a high-level page description language which could be used to describe the appearance of any page (containing text, graphics and digitised images) in a device independent way. Both Warnock and Geschke were able to draw upon the experience gained in their work at Xerox, where they contributed to the development of the Interpress page description language, an early but now faded rival to PostScript.

Adobe's first big commercial breakthrough came in 1985 when Apple launched the first PostScript printer, the Apple LaserWriter. The LaserWriter incorporated a PostScript interpreter which was capable of executing a PostScript program and of generating the printed pages which it described. It's often said that the 'DTP revolution' was brought about largely through the efforts of four companies: Apple, with the Apple Macintosh; Aldus, with PageMaker; Canon, with cheap laser printing technology; and last but not least, Adobe, with PostScript.

Since then, PostScript has grown to become the *de facto* industry standard page description language, with PostScript interpreters implemented on low-end office laser printers and high-end typesetting machines alike. Adobe now boasts revenues in excess of $150 million per year, from licensing its PostScript interpreter and from PostScript based applications programs, such as Adobe Illustrator and PhotoShop.

Most users will be blissfully unaware of PostScript since PostScript programs (or document descriptions) are normally machine generated. However, it is a powerful interpretive programming language in its own right, and so warrants inclusion in .EXE's series of articles on 'The Third Side'.

PostScript is interpreted, stack-based, and extensible via its powerful 'dictionary' data structure

The PostScript Language

The three main features of PostScript which distinguish it from most other programming languages are that it is interpreted, it is stack based, and it is extensible through using its powerful 'dictionary' data structure.

As well as all the standard arithmetic and control operators, PostScript also includes a powerful set of graphics operators for drawing arbitrary shapes which can subsequently be either filled or outlined ('stroked', in PostScript terminology) using a range of different attributes such as pen width and fill pattern. PostScript provides operators for rendering bitmap image data on a page, and also contains a set of operators for controlling its sophisticated outline font scaling technology so that high quality text may be intermingled with other graphical objects. Another way of thinking about this is that within PostScript, a 'character' is just another graphics object: its shape is specified algorithmically.

The Interpreter

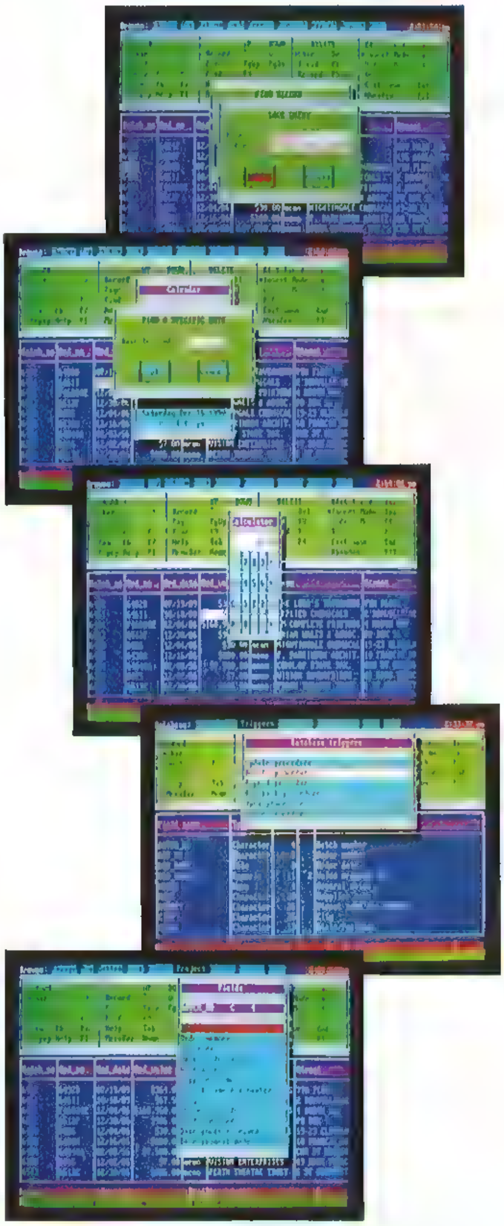
To understand PostScript programming, it is important first to have an idea of how PostScript interpreters work.

The PostScript programming language is designed to be interpreted, not compiled, and the PostScript interpreter 'consumes' the PostScript program from its input stream as it is needed, executing the program 'on the fly'. There is no requirement for reading in the text of the program prior to executing it, unlike many other pre-compilable programming languages. An important aspect of this is that the PostScript interpreter does not impose a structure on the PostScript program. However, Adobe strongly recommends that all PostScript programs conform to the structure described briefly below.

Most PostScript interpreters may operate in one of two modes, either *batch* mode or *interactive* mode. By far the most common usage is for the interpreter to work in batch mode, where a PostScript program is prepared by an application and transmitted to the PostScript interpreter. Although there may be some dialogue between the application and the interpreter (for example, where the application queries the printer to find out which fonts are available) this is invisible to the user. Error messages will be sent back from the PostScript interpreter to the host application which may display them on screen to the user.

In interpretive mode, a person may effectively 'log in' to the PostScript interpreter and establish a dialogue with it (usually by typing in the `executive` command). The PostScript interpreter will then respond with a prompt and will normally allow some degree of in-line editing. This mode is really only used for experimenting with

Recital 7.0 Is The Only Choice.



- Advanced RDBMS/4GL
- Active Data Dictionary
- Object-oriented with pre-defined "application objects"
- Open client/server architecture
- Integrated pop-ups for calendar, calculator, choice lists
- Built in "Dynamic Compiler" New v7.0
- Pop-up time management scheduler New v7.0
- Relational multi-record table fields in forms New v7.0
- Split screen relational browse New v7.0
- Database, form and report triggers New v7.0
- Automatic record locking
- Supports 14 datatypes including memo, numeric, byte, integer, short, character, date, logical, float, real, packed, zoned, vaxdate, quadword
- Available now on 100+ VAX/VMS and UNIX computers (including full VAX cluster support)
- SQL (interactive and embedded)
- dBASE, FoxBase and Clipper compatibility (yes, even on VMS!)
- Terminal (including full color support) and keyboard independence
- Transparent connectivity to existing RMS and RDB databases on VAX/VMS
- Completely integrated development environment
- Powerful end-user tools

Recital
CORPORATION

Rdb RSVP
SOLUTIONS • VENDOR • PROGRAM

COMPATIBILITY 88open CERTIFIED

Recital Corporation LTD, South Bank Technopark, 90 London Road, London SE1 6LN, United Kingdom, Tel 071-401-2727, Fax 071-633-9617
Recital Corporation Inc., 85 Constitution Lane, Danvers, MA 01923, Tel (508) 750-1066, Fax (508) 750-8097

CIRCLE NO. 885

Security Software from Microft Technology.

MENUGEN

MENUGEN provides a password protected menu interface. The system manager has complete control of what each user is permitted to do. Access to the DOS prompt can be prevented. MENUGEN has no memory overhead.

D-LOCK

D-LOCK prevents access to the hard disk after booting from diskette.

CLAM

CLAM provides TOTAL security. It incorporates MENUGEN and D-LOCK. In addition it provides a number of other facilities, the most important of which is file encryption – the only way to provide a high level of security on a micro.

Microft Technology Limited
The Old Powerhouse, Kew Gardens Station
Kew, Surrey TW9 3PS.
Telephone: 081-948 8255

CLAM £148+VAT Single User.
MENUGEN £48+VAT Single User.
D-LOCK £38+VAT Single User.
Corporate Licences available.

CIRCLE NO. 886

the language, for example by entering the triangle program listed in Figure 4.

In both cases, the PostScript interpreter executes the PostScript code continuously from the time it receives the initial byte, via the communications channel, to the next end-of-file (EOF) indication. The EOF indication is not a syntactic element within the PostScript language, rather it is part of a higher level structure imposed on a continuous input stream to structure it into individual *jobs*. A vital part of this higher level job structure is that at the end of the job, the printer resets itself to the state which existed at the beginning of the job. This means that all changes made to the state of the interpreter (eg defining new procedures and variables) will be lost at the end of a job.

Program structure

Unlike many other programming languages, there is no concept of a program entity within the PostScript language and no requirement for the PostScript program to be read in its entirety before beginning execution.

By convention, however, Adobe recommends that a PostScript program be divided into two parts: the *prolog* and the *script*. The prolog contains the definitions which are to be used in the script, the script contains the executable part of the PostScript program. In addition, Adobe is very keen for developers to use its *document structuring* conventions, which use special comments to impose a sort of structure on a PostScript program.

In most circumstances, the script will contain individual page descriptions which are described using a combination of the built-in operators and new procedures which are defined within the prolog. The convention here is that all page descriptions are independent of each other, and dependent only on the procedures and definitions within the prolog. This allows printer management software to extract individual pages from a PostScript program, using information contained in the document structuring comments.

Figure 1 gives an example of a simple PostScript program which uses some of the Adobe document structuring comments and shows some of the basic graphics operators in action, such as `lineto`, `moveto` and `fill`.

The prolog for the program contains a single definition for a `square` procedure which constructs the outline of a square of 72 points by 72 units (the default PostScript coördinate system is in points, where 72 points = 1 inch). The script contains two page descriptions which use some of the built-in graphics operators together with the `square` procedure defined within the prolog.

When this program is sent to a PostScript printer it will first print a page with a black filled square, and a second page with an outlined square. Both squares will be positioned at the bottom left hand corner of the page (0,0).

Syntax

Perhaps the most striking (and initially confusing) feature of a PostScript program is that it uses a postfix notation in which operators are preceded by their operands. In this respect, it is similar to the programming language FORTH. A simple example of this is given in the `square` program listed in Figure 1, where all coördinates to the `moveto` and `lineto` operators precede the operator itself.

As a more complicated example, in the triangle program listed in Figure 4, the segment:

```
sidesarray 0 get sidesarray 1
get sidesarray 2 get add gt
```

is equivalent to the 'C' expression:

```
sidesarray[0] >
       (sidesarray[1] +
               sidesarray[2])
```

Stacks

The postfix notation goes hand in hand with the stack based, interpretive nature of the PostScript language.

The PostScript interpreter maintains four stacks, the *operand*, *graphics*, *dictionary* and *execution* stacks, all of which are Last In First Out (LIFO) stacks. The operand stack is the one which we are most concerned with here, and is used to hold the operands and results of PostScript operators being executed. It is usually referred to just as 'the stack'.

Consider the `add` PostScript operator, which removes the top two numbers from the operand stack, adds them together, and pushes the result back onto the operand stack. Figure 2 shows how the operand stack grows and shrinks during the execution of the expression:

```
1 2 3 add add
```

This can make PostScript programs difficult to understand until you get used to the idea of thinking in terms of stacks. For example, there is nothing to stop someone writing a program which puts an object on the operand stack several hundreds of lines before that object is used. In most other programming languages, it is immediately clear which operand goes with which operator by its context within the program text.

```
%!PS-Adobe-2.0
%%Title: box.p
%%Creator: Corinna Kinchin
%%CreationDate: 8 Apr 1991
%%For: .EXE
%%BoundingBox: 0 0 595 842
%%Pages: 2
%%DocumentFonts: None
%%EndComments

% ---         square => -
%             construct square 72 x 72 points
/square
 {
              0 0 moveto
              72 0 lineto
              72 72 lineto
              0 72 lineto
              closepath
 } def

%%EndProlog

%%Page: 1
save             % --- save current state
square           % --- construct square outline
0 setgray        % --- select black ink
fill             % --- fill square
restore          % --- restore to saved state

showpage         % --- print page

%%Page: 2
save             % --- save current state
square           % --- construct square outline
1 setlinewidth   % --- pen width=1 point
stroke           % --- stroke square
restore          % --- restore to saved state

showpage         % --- print page
```

Figure 1 - PostScript program for printing squares

Data types

The complete list of object types supported by PostScript is: `integer`, `real`, `boolean`, `array`, `packedarray`, `string`, `name`, `dictionary`, `operator`, `file`, `mark`,`null`, `save` and `fontID`. Most of these are simple *atomic* entities, ie they contain no visible substructure. Objects of types `array`, `packedarray`, `dictionary` and `string` are composite objects, ie they have an internal substructure. Of the composite objects, the most unusual is the `dictionary` object which will be covered later on.

Operators

PostScript has all the standard arithmetic and control operators, as well as a comprehensive set of graphics and font related operators for handling the typesetting and rendering of page descriptions.

Although PostScript contains over 250 built-in operators, there are no reserved words in the language. In fact, it is possible to redefine any of the built-in operators so that they do something quite different to what may have been originally intended.

For example,

```
/add { sub } def
```

will redefine the PostScript `add` operator to subtract rather than add the top two numbers on the operand stack. Although this seems dangerous, the feature can be (and is) often used to great effect. A common application of this is to redefine the built-in PostScript `showpage` operator to add crop marks to every page printed while that redefinition is in effect.

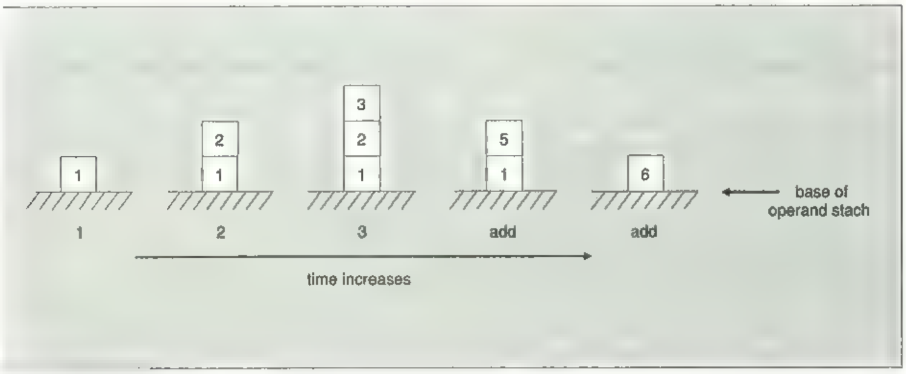


Figure 2 - The PostScript operand stack

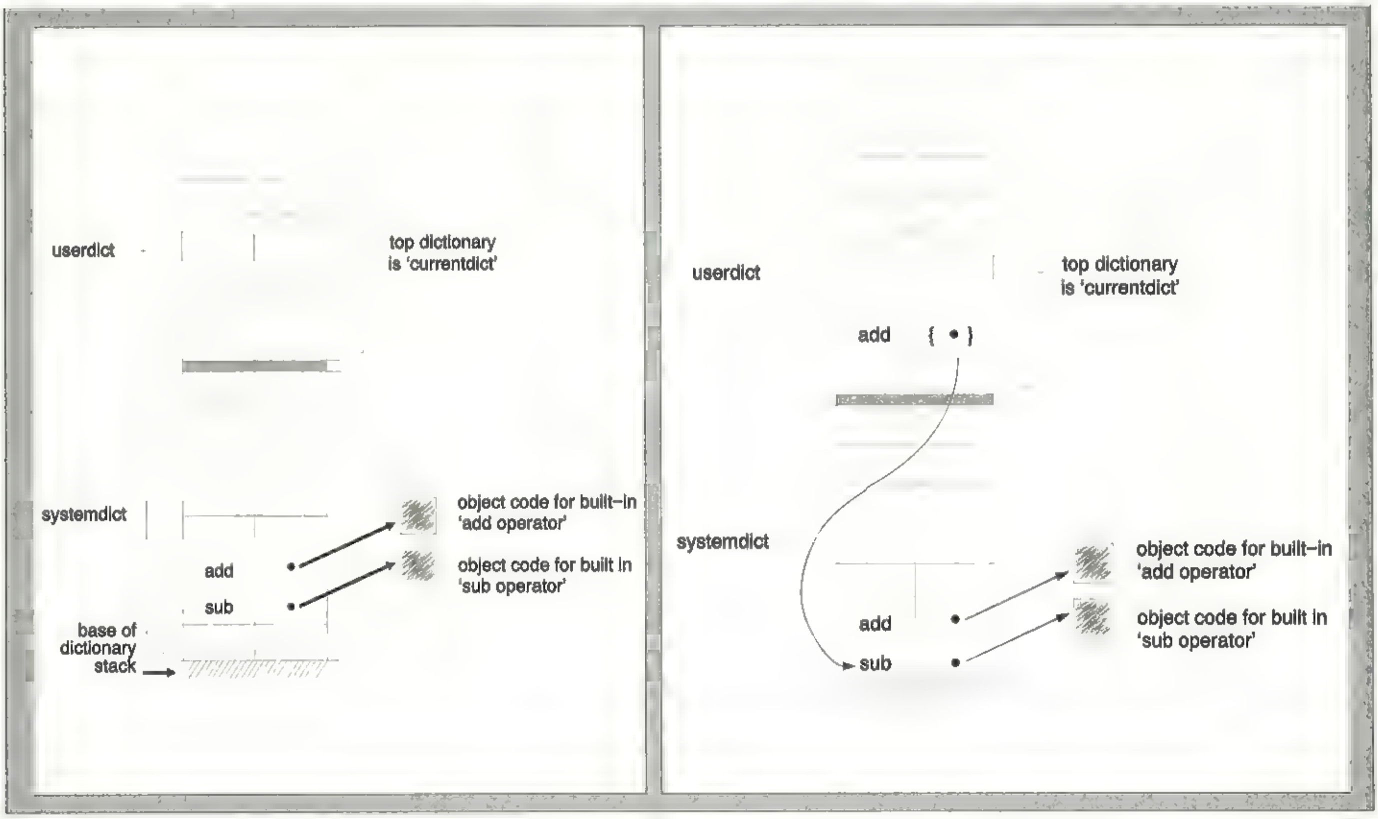
Dictionaries

This brings us on to a powerful feature of PostScript which is that of *dictionaries*. A dictionary is a composite data structure with room for a number of key-value pairs, where the key is a PostScript *name* (eg the name of an operator or a procedure) and its value can be of any type (including object code).

The PostScript language provides a range of operators for creating and modifying dictionary contents as necessary. Dictionaries provide a way of temporarily extending the PostScript language and are the only way to store PostScript objects for later recall.

As mentioned earlier, the PostScript interpreter maintains a dictionary stack. The bottommost dictionary on the dictionary stack is always `systemdict`, and at the start of a PostScript program the topmost dictionary is an (almost) empty dictionary called `userdict`. The topmost dictionary on the stack at any point during the execution of a PostScript program is known as the *current dictionary* (see Figure 3a).

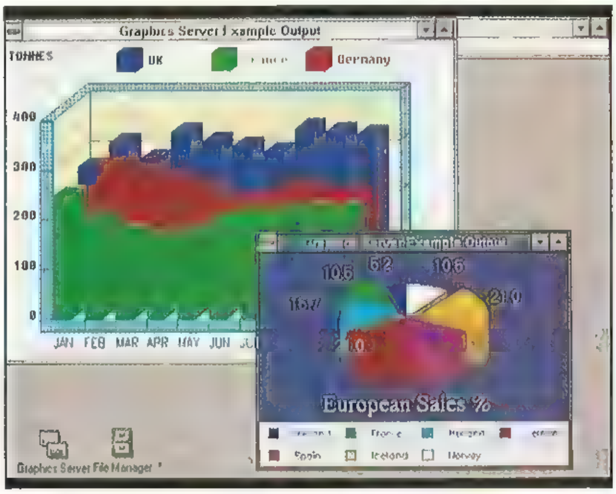
Almost all of the built-in PostScript operators (such as `add` and `def`) are simply keys whose values (the executable part) are defined in the `systemdict` dictionary, which is a standard data structure maintained within all PostScript interpreters.

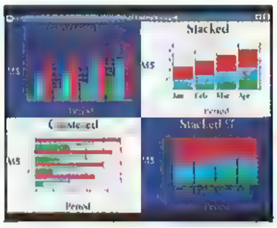


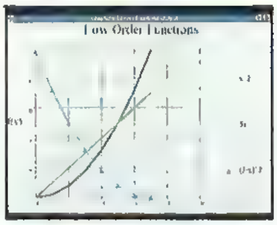
Figures 3a/3b - The PostScript dictionary stack

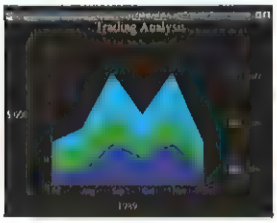
Bring Graphs and Charts to your Windows Applications Licence-Free with Graphics Server™

Introducing the **Graphics Server Software Development Kit (SDK)** from Bits Per Second. The most cost-effective, time-efficient way to integrate full-colour graphs and charts into your Windows applications. Here's why.









Client-Server Model

Graphics Server is a Windows server process which responds to multiple, concurrent requests for interactive graphing and charting functions from client applications. It enables graphs and charts to be incorporated into any Windows application having a Dynamic Link Library (DLL) or Dynamic Data Exchange (DDE) capability. This includes C programs, Superbase, SQL Windows, ToolBook, Word for Windows and many others. In fact, virtually all Windows applications!

Multiple Choice

Graphics Server lets you choose from 16 graph types, including 2-D and 3-D bar (simple, stacked and clustered), 2-D and 3-D pie, area, Gantt, scatter, polar, log-log/log-linear, line, high-low-close, X-Y and polar, plus a variety of statistical functions.

Ease of Programming

Graphics Server, through its multi-level interface, provides the level of control **you** want. In as few as **eight** calls you can open a new graph window, define titles, legends and labels, transfer data, draw a chart, and output the chart to the clipboard or printer. Graphics Server takes care of all the details; scaling the graph to match the data, matching fonts, text size and position to the graph, even resizing and repainting the graph window when necessary! You'll save development time. And money.

Alternatively, if you want full control yourself, Graphics Server provides over a **hundred** functions to help you design, create and manage virtually any kind of graphically drawn image. Multiple graphs, multiple windows, user interaction via hot regions. All these and more!

Full Support

The Graphics Server SDK gives you all the tools, documentation and examples you'll need to make immediate and effective use of Graphics Server. Within hours your applications will be transformed by the inclusion of presentation-quality graphics.

Telephone/fax support and a bulletin board service are provided for all registered users, and for power users requiring an even higher level of service and security there's a special VIP Support Programme.

No Licences!

The Graphics Server SDK costs £275 + VAT. No further licence fees to pay.

To place an order, request a demo disk, or obtain additional information, call . . .

BITS PER SECOND

Bits Per Second Ltd
14 Regent Hill
Brighton BN1 3ED
Tel (0273) 727119
Fax (0273) 731925

All product names are trademarks of their respective holders.

CIRCLE NO. 887

Desktop FORTRAN

EXPERTS AGREE WHAT TO BUY – THEY JUST CAN'T AGREE WHY!

"The . . . FTN77 compiler has allowed us to port our main-frame applications to 386 micros with minimum effort. The speed of compilation is incredible!. . . The run time performance is fantastic, full support for 32-bit operations and the 80387 coprocessor means our programs run like lightning!"
Tony Fitzpatrick, Exploration Consultants Ltd.

"I find the FTN77/386 compiler to be fast and easy to use . . . . We often download programs under development from a mainframe to the PC to run through this computer."
Rob McLaren
FTN77/386 User.

"Without the FTN77/386 compiler it would have been totally impractical for us to convert our million-line + power system analysis program (PSS/E) to run on a 386 PC."
Wayne B'Rells
Power Technologies Inc.

"FTN77/386 provides more intrinsic functions than any other functions to manipulate strings, handle screen and keyboard I/O, read the date and time, parse the command line . . . Advanced file manipulation functions allow an FTN77/386 program to work with files in a C-like manner".
. . Scott Ladd,
Computer Language, Nov. 1988.

". . . I am very impressed and have decided to shift all our production programs over to FTN77. . ."
Dr. G.R. Chaplain
NCL Investments Ltd.

"The ease of use, speed, debugging facilities, dynamic libraries with resulting small .EXE files, and extensive library routines made it a joy to use."
H. Gordon Jensen
FTN77/386 User.

"By far the fastest and most complete FORTRAN compiler that I have ever seen or used in the DOS environment".
Tim Wool
FTN77/386 User.

"The debugger was very easy to use. There is an on-line help facility which made reference to the manual unnecessary."
. . . Mike Gunn and Arul Britto,
EXE Magazine, May 1989.

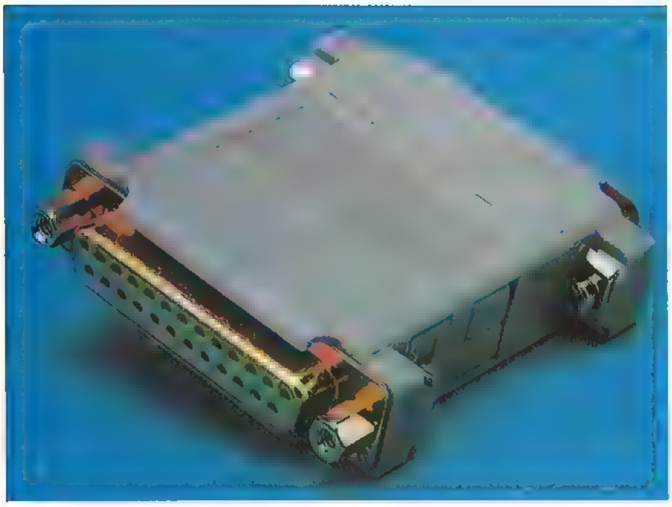
"For sheer speed and sleekness the (Salford) compiler stands out like a Ferrari among Yugos"
. . Jack W. Crenshaw, Computer Language, May 1990

FAST, FRIENDLY & FULLY FEATURED – THE PERFECT MIX – CALL US NOW!

SALFORD SOFTWARE MARKETING LTD. MAXWELL BUILDING · THE CRESCENT · SALFORD M5 4WT.
TELEPHONE: 061 745 5678 · FAX: 061 745 5666
FTN77 386/486 – extending MS-DOS.

UNIX VERSION AVAILABLE NOW!

CIRCLE NO. 888



How many users of your software paid for it???

SOFTLoK International Limited was established in 1987 with the introduction of our SOFTLoK and SOFTLoK PLUS devices to combat the ever increasing problem of software piracy. Our SOFTLoK range of software protection devices are used by hundreds of software developers from small consultancies to large multinationals. To cope with the ever changing needs of our customers we have developed SOFTLoK II which combines the programmable features of SOFTLoK PLUS with a low unit cost similar to our original SOFTLoK product.

SOFTLoK II™

The Next Generation

SOFTLoK II units are programmable devices containing read/write memory protected by a password. Both the memory and the password can be changed at any time using our routines in your application software. Easy to use menu-driven software is provided to allow small or large batches of SOFTLoK II units to be programmed with their initial data & passwords ready to be sent out with the protected software product.

Price: 1-19 £16.90, 20-49 £15.40,
50+ £13.70

Evaluation kit £20 (SOFTLoK II, manual & software)
All prices excl. VAT and delivery

As SOFTLoK II units plug into the parallel printer port they can be installed or removed in seconds.

□ For IBM PC, PS/2 and compatibles
□ Uses parallel printer port
□ Totally transparent to printer
□ Secure data & password can be changed from your application software
□ Cascadeable
□ 240 bytes of secure read/write memory
□ 8 byte (64 bit) password
□ No programming adaptors required
□ Easy to use SOFTLoK II setup software
□ Routines ready to link with various compilers
□ Easy to follow manual

I-MEX House, 40 Princess Street,
Manchester, M1 6DE, England,
Tel: 061 228 7379 **Fax:** 061 236 6890

CIRCLE NO. 889

Presenting a key to the PostScript interpreter causes it to perform a name lookup on that key and to execute its corresponding value. (If it can't find that name, the interpreter will generate an error message and the rest of the current job will be discarded).

In the case of the built-in `add` operator, the value associated with it in `systemdict` is the object code necessary to remove two items from the stack, add them together, and push the result back onto the stack. Thus, when the interpreter encounters the `add` operator it looks up the 'add' name in `systemdict` and executes the value associated with it (in this case, some object code).

The `systemdict` dictionary is illustrated in Figure 3a, showing the `add` and `sub` operators with their associated object code. This is a schematic diagram showing the object code stored elsewhere in printer RAM.

Now let's take the example of redefining the `add` operator using:

```
/add { sub } def
```

Here, `/add` is a key and `{ sub }` is its value. The `def` operator takes these two objects (from the top of the operand stack) and inserts them as a key-value pair into the current dictionary (in this example we will assume it to be `userdict`). Figure 3b shows the state of the dictionary stack immediately after the execution of the `def` operator.

The same key may be defined in several different dictionaries, but the PostScript interpreter will use the one it finds first during its top-down search of the dictionary stack.

In this example, when the interpreter encounters an `add` name in the input stream subsequent to this redefinition, it will perform a double lookup, the first to look up `add` and the second to look up `sub`. Finally, it will execute the value of `sub`, in this case some object code to subtract the top two items on the stack and to push back the resulting value.

But what if you wanted to use the original value of the `add` operator? No panic. PostScript includes a number of built-in operators for extracting a value from a particular dictionary. For example, the following code fragment extracts the original built-in value of the `add` operator and executes it, to add the top two numbers from the operand stack and to push back the result:

```
1 1 systemdict /add get exec
```

Immediate vs deferred

So far in this article the implication has been that the PostScript interpreter scans in objects from the input stream and executes them immediately. In the case of PostScript procedures, however, this is not strictly true.

PostScript procedures are delimited with curly brackets '{' and '}'. When the interpreter encounters the opening curly bracket '{' it stops execution until it finds a matching closed curly bracket '}'. All the objects contained within the curly brackets are assembled into a single procedure object suitable for later recall and execution. This is known as *deferred execution.*

Another case of deferred execution is that of *literal names* vs *executable names.* In the example of redefining the `add` operator using:

```
/add { sub } def
```

the use of the character / immediately before the `add` operator prevents it from being executed immediately, thus allowing its definition to be modified. Here, `/add` is a literal name, whereas `sub` (without a preceding / character) is an executable name. However, since it occurs within a procedure body it is not executed immediately but is stored within a procedure object for later recall.

The Triangle Program

The triangle program listed in Figure 4 is an unusual, and not very useful, form of a PostScript program, since it must be entered in interactive mode as explained earlier.

The PostScript interpreter expects to read the program source from the same input

```
%!PS-Adobe-2.0
%%Title: triangle.p
%%Creator: Corinna Kinchin
%%CreationDate: Mar 1991
%%For: .EXE magazine
%%BoundingBox: 0 0 595 792
%%EndComments

% ------------------------------------------
%       Variables
% ------------------------------------------

/SIDES_PER_TRIANGLE 3 def

% ---   Array of side lengths
/sidesarray SIDES_PER_TRIANGLE array def

% ---   String to accept number
/numstr 10 string def

% ---   open standard input for reading
/infile (%stdin) (r) file def

% ------------------------------------------
%       Procedures
% ------------------------------------------

/Triangle
 {
        % ---   read three side lengths into sidesarray
        0 1 SIDES_PER_TRIANGLE 1 sub
         {      % for
                % ---   stack contains: index

                % ---   read side length from input stream
                (\nEnter length of side: ) print flush
                infile numstr readline
                { cvi } if              % --- pop off bool value and convert string to int
                % ---   stack contains: counter value
                % ---   put value into array at offset
                %       given by current value of counter
                sidesarray 3 1 roll put
         } for

        % ---   Print numbers stored in array
        (\nTriangle entered: ) print flush
        sidesarray { == } forall

        % ---   Check triangle inequality
        sidesarray 0 get sidesarray 1 get sidesarray 2 get add gt
        sidesarray 1 get sidesarray 2 get sidesarray 0 get add gt or
        sidesarray 2 get sidesarray 0 get sidesarray 1 get add gt or
         { (This is not a triangle.\n) print flush }
         {      % else
                % ---   Check whether equilateral
                sidesarray 0 get sidesarray 1 get eq
                sidesarray 1 get sidesarray 2 get eq and
                sidesarray 2 get sidesarray 0 get eq and
                 {      (Triangle is equilateral.\n) print flush }
                 {      % else
                        % ---           Check whether isosceles
                        sidesarray 0 get sidesarray 1 get eq
                        sidesarray 1 get sidesarray 2 get eq or
                        sidesarray 2 get sidesarray 0 get eq or
                         {              (Triangle is isosceles.\n) print flush }
                         {              % else
                                        (Triangle is scalene.\n) print flush
                         } ifelse
                 } ifelse
         } ifelse

 } def
%%EndProlog

Triangle
```

Figure 4 - The Triangle program

stream as the data to be used within the program. This means that to run the Triangle program you must first establish an interactive dialogue with the PostScript interpreter and type in the full text of the prolog in which the triangle procedure is defined.

To execute the triangle procedure, you then type:

```
Triangle
```

The Triangle program will prompt you to enter the lengths of three sides of the triangle and the program will respond appropriately.

The definition of the Triangle procedure will remain available to you until the end of the current job, ie until you enter the EOF character (Ctrl-D on many systems), so you can execute it again by retyping `Triangle`.

Level 2

There is now a new version of PostScript called 'PostScript Level 2', which contains substantial additions and enhancements to the standard language covered in this article. Adobe has given a commitment that Level 2 interpreters will be backwards compatible with the standard version defined in the original PostScript Language Reference Manual (the Adobe 'Red Book').

Changes in the language include better and faster text rendering; enhanced colour capabilities; and built-in data compression and decompression algorithms.

The first Level 2 PostScript printer (courtesy of DataProducts) was due out in May 1991.

Conclusion

This article is unusual in that it has centred around using PostScript as a programming language rather than its use as an imaging model. In real-life, PostScript is used almost entirely for its merits as a standard page description language, and as much of the processing as possible should be done on the host computer which submits the PostScript job, rather than on the PostScript interpreter itself. Poorly written PostScript programs can tie up a PostScript printer for hours.

But it's a great language to experiment with, if only to see what spectacular results can be produced on your office laser printer.

Go for it!

EXE

Corinna Kinchin is a contributing editor for the PostScript Review *(formerly the* PostScript Language Journal - International Edition*), a quarterly magazine devoted entirely to PostScript and related products. It contains news, reviews, technical tips and PostScript reference listings, and is available by subscription from: The PostScript Review, 133 Notting Hill Gate, London W11 3LB (Tel 071-792 9429)(Fax.071-727 6045)*

For further reference, Corinna suggests PostScript Language Reference Manual *(the 'Red Book') by Adobe Systems Inc, pub Addison-Wesley, ISBN: 0-201-10169-6;* PostScript Language Tutorial and Cookbook *(the 'Blue Book') by Adobe Systems Inc, pub Addison Wesley, ISBN: 0-201-10189-0;* PostScript Language Program Design *(the 'Green Book') Glenn C. Reid, Adobe Systems Inc, pub Addison-Wesley, ISBN: 0-201-14396-8;* Real World PostScript *ed by Stephen F. Roth, pub Addison-Wesley ISBN: 0-201-06663-7; and, of course, her own journal* The PostScript Review.

Five New Bundesländer

When East Germany joined the West, there were some unfortunate consequences for the local computing community. W W Osterhage explains.

The walls came tumbling down, and CoCom lost a piece of its puzzle. On November 8th 1989, the Berlin Wall lost its significance, during early 1990 there was economic union and, in Autumn 1990, East and West Germany united bringing FNB - 'Five New Bundesländer' - to the West. Two-way access was created: the East Germans got Western high technology; the Western high technology suppliers got the East Germans. And the most important component of high tech, as readers of this magazine must know, is computer hardware and packaged software.

Before union, access to computers had been restricted in two ways: technologically and administratively. The technology was restricted to Soviet built mainframes, Bulgarian disk drives and East German desk-top machines. On the administrative side, computer installations were kept under lock and key (just like telex machines), and operating and management personnel were in 'close liaison' with the East German equivalent of the KGB: Stasi. (I have this from someone directly concerned.)

Desk-top equipment was made by the large kombinat 'Robotron'. The technology trailed the West by some five years, and most of the equipment went for export. Although East German TV was constantly pumping out word processing courses, the general public hardly ever saw the Robotron machines: there were too few and they cost too much. Some adventurous and/or better off people made it to IBM PCs or clones. The rest starved.

So after the political events the market was cracked wide open. Many a private citizen travelled to Western distributors to spend some of his or her savings - converted to Deutsch Marks - on computing machinery; while many a salesman, preceded or followed by a very mixed assortment of consultants, travelled east to peddle hardware and software - mainly hardware. I was (am) one of those consultants. I was shocked by the situation in the field, which I got to know first hand by talking to a number of managers in the FNB. What was once the inaccessible near-East had become the free-for-all far-West.

Initially most of the several thousand East German companies, and especially the kombinat, still had money to spend. It was at this point that greedy mainframers came in and made quick contracts. Then the Treuhand, as collateral administrator, took over all state property (its late boss - a Mr Rohwedder - was assassinated in April by terrorists) and started salvaging by cutting the work force. At the same time, the old East European clients were lost, because they could not pay the new Westerners in hard currency. These developments reduced the remaining ambitions of Eastern managers for computerised solutions in their modernisation drive.

Bad Examples

But they were and are lost because of a lack of expertise, experienced judgement and total dependence on an aggressive sales force. To illustrate this dilemma I would like to describe two cases, brought to me by desperate men in search of ways out.

The manager of the DP department of a medium-sized tool machine factory (4000 employees) was approached by a minor-league (below the top five) mainframe manufacturer, who persuaded him that his ailing company needed a production planning and control system. They persuaded him to turn his department into a profit centre (80 people, out of which 15 were hardware maintenance engineers required to keep the Russian hardware going), if he would let the mainframer permanently on site to provide demos for prospective clients. The DP department was to receive a 5% cut of any successful sale. This seemed a reasonable deal, so the manager bought the cost control package.

One month later, the same man was approached by another, bigger, mainframe manufacturer, who told him they had all the solutions and a big machine. This supplier was backed by the new (now Western) Governing Board of the factory.

At the same time, one of the manager's new sister companies from the West sent in its experts, suggesting he needed a French mainframe, with French software and documentation.

The DP manager could not sort out this three-way political pressure or resolve his contractual obligations. So he called in independent consultants to sort out the mess. That was when he ran out of money to pay them.

Another story. An electrical equipment maker (kombinat) decided that it needed a production planning and control system. Without establishing specifications, it bought a PC-based off-the-shelf software package from an unknown West German supplier. Then the kombinat was broken up, and its various production centres sold off. The software remained with one of the centres, which now faces a double dilemma: does the remainder still need it? and does it fit with the DP strategy of the new mother company?

There is work to do in the East. There was a lot of work to do at the start. Now, after the first wave of Western technology, there is still more work. Tasks, which in the absence of historically grown systems, could for once have been technically straightforward, have now reached, in places, the same level of complexity as the maintenance of Western applications. All this achieved in a few short months.

There are still opportunities for serious people to help and teach - more so now than ever - but there is less and less money.

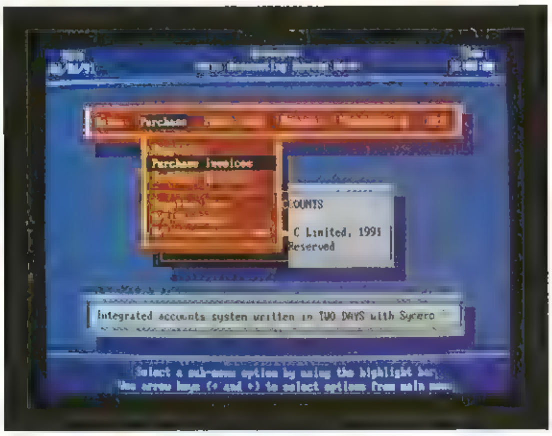
EXE

Wolfgang W. Osterhage holds PhDs in Physics and Information Science. He works as an independent consultant for industry, specialising in IT and logistics. He can be contacted at Birkenweg 7, D-5307 Wachtberg - Niederbachem, Germany.

SEE US ON
STAND 35 IN
EXHIBITION
HALL 1 AT
SOFTWARE
TOOLS '91

SoftwareDevelopers!

Produce a sophisticated application like this in Microsoft C , with either BTRIEVE or C-ISAM, running under DOS in just a couple of days!



Purchase Invoices

COUNTS

C Limited, 1991
Reserved

Integrated accounts system written in TWO DAYS with Sycero

Select a sub-menu option by using the highlight bar,
Use arrow keys (← and →) to select options from main menu

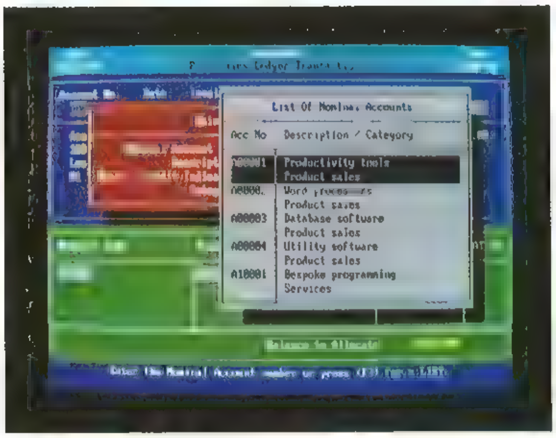
...Here's the same application half an hour later usingTurbo C ++'s advanced VROOM technology...

Enter the nominal code, press (F3) for a LIST, (F7) - ...

...Another twenty minutes and it can access dBASE files...

Enter the Nominal account number or press <F3> for a LIST

...and here's the same application an hour later running under SCO UNIX, with no runtime licences required !



List Of Nominal Accounts

Acc No	Description / Category
A00001	Productivity tools Product sales
A0000.	Word processors Product sales
A00003	Database software Product sales
A00004	Utility software Product sales
A10001	Bespoke programming Services

Enter the Nominal Account number or press (F3) for a LIST

This entire application was developed in a couple of days using **Sycero C**. Sycero is a powerful program generator that **dramatically** reduces your programming time. Sycero C generates fully structured and documented C source code to be compiled with Microsoft or Turbo C. Sycero C supports the Btrieve/C-ISAM file handling systems, plus it can even generate C programs for use with dBASE or Clipper files and indexes. Sycero C UNIX allows you to develop software to run under SCO UNIX, and you can regenerate existing Sycero C DOS-based applications with absolutely **no changes** required to your definitions.

For more information, either call us on **0622 691616**, or fill out and return the coupon.

CIRCLE NO. 891

dBASE is a trademark of Ashton-Tate. All other trademarks acknowledged.

Please send me further details about Sycero C

Name: ....

Company: ....

Address: ....

....

....

....

.EXE 6/91

Tel: .... Fax: ....

I am also interested in your generator for Clipper, Sycero dB □

System C Ltd 60-61 High Street Maidstone Kent ME14 1SR

Tel 0622 691616 Fax 0622 691241

For total portability right now, your choices are a bit limited.

Only Zortech Gives You Total Portability to MS-DOS, Windows, OS/2, DOS 386, UNIX 386. And Macintosh. Right Now.

What Platform Will Your Application Be Running On This Time Next Year?

Now that C++ has made significant inroads into mainstream application development, you must choose the C++ compiler that is going to be the performance and portability leader now...and years to come. Since 1986, we've been testing, refining and perfecting our family of C++ products tirelessly. The results are what you are reading about, a fully portable family of C and C++ products available right now.

BUYERS ASSURANCE SEAL
INFO WORLD

We Intend To Keep Up This Pace.

In the past few years, Zortech has quietly become an industry standard earning acclaim from Byte, Computer Language, Dr. Dobbs, and many industry experts for our solid implementation, a market-leading performance, and strong, relentless innovation.

Whatever Platform You Choose, Choose Zortech Performance.

With Zortech, no matter what platform you choose to develop on, a substantial performance increase will be realized due to the quality of the original Zortech C++ implementation. For instance, on MS-DOS, your program will compile up to 35% faster with the resultant code running up to 45% faster and 25% smaller than Turbo C++*. But we don't stop there. On each platform, Zortech C++ is designed to improve your productivity where it really matters... in the development cycle.

MICROSOFT WINDOWS

For example, using the DOS/MS Windows package, you can edit, compile, link and debug the largest MS windows applications...and never once leave the Windows environment. All this plus complete plug-and-go to CNS C++/View and Commonview II! And, of course, these advantages are also available on the OS/2, DOS 386, UNIX, and Macintosh platforms.

60 DAY MONEY-BACK GUARANTEE

Choices. Choices. Choices...

If there is one constant in C++ development, it's change. The Zortech family of products keeps your options exactly how they should be. Open.

To find out more, call today.

Order Hotline:

44-81-316 7777

Zortech

Industry Standard. Industrial Strength.

Zortech, Inc., 4-C Gill Street, Woburn, MA 01801 Tel: (617) 937-0696 Fax: (617) 937-0793
Zortech, Ltd., 58-60 Berkeford Street, London SE18 6BG Tel: +44-81-316-7777 Fax: +44-81-316-4138
Supported systems include Compaq, ISA/EISA, IBM PS/2 and 100% compatibles

* Source: MS-DOS, C++ Compilers: A comparison by the Ladd Group. (c) 1990

CIRCLE NO. 892

ZipitUP™

PC SOFTWARE PROTECTION SYSTEM

- Totally Protects MSDOS Programs from Piracy, Illicit Copying and Unauthorised Use
- Easy to Use – No programming – Software only
- Unbeaten Record of Applications to Products Throughout the Computer Industry Worldwide
- U.K. Developed and Supported

ZipitUP Gives You All of This –

* Very High Protection Levels * Totally User-Transparent and Friendly
* Runs on All 3.5" and 5 25" LD and HD formats
* No Expensive Add ons *Use *ZipitUP* on your PC, PC Disk Autoloader or have your Duplicator Bulk Copy it for you

ZipitUP Core Utilities – £99.00 + vat

Includes Everything to Protect Programs onto *ZipitUP* Key Disks for Key Disk and Hard Disk Operation

PLUS 4 FREE Sample *ZipitUP* Key Disks and the Complete *ZipitUP* Reference Manual for all *ZipitUP* Utilities and Extra Modules

ZipitUP Evaluation Pack – £25 inc vat/UK pp

This Unique Offer provides 3 sample opportunities to protect your Software is supplied with Full Documentation. The cost will be Credited against subsequent purchase of *ZipitUP* utilities.

FREE ZipitUP Demo available. For Further Information and Brochure contact us.

"See Us On Stand 33 Software Tools '91"

THE data BUSINESS *Experts in Data Resources*

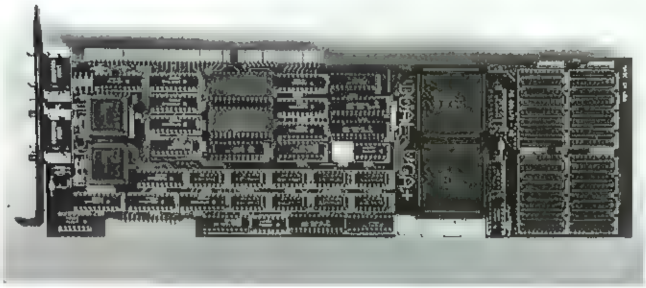
The Data Business Ltd,
3 Bankside, Kidlington, Oxford OX5 1JF, UK
Telephone: (0865) 842224 Fax: (0865) 841507

ZipitUP the Best Answer There Is...

CIRCLE NO. 893

MULTIPLE INDEPENDENT MONITORS
CAN NOW BE DRIVEN SIMULTANEOUSLY
FROM A SINGLE PC OR PS/2

THE DUAL VGA PLUS



★ UP TO 800×600 RESOLUTION

★ TWO INDEPENDENT SIMULTANEOUS DISPLAYS WITH OVERLAY CAPABILITY

★ 100% COMPATIBLE WITH IBM VGA/EGA/CGA/MDA VIDEO STANDARDS

★ 4 BOARDS IN ONE SYSTEM GIVE 8 SIMULTANEOUS DISPLAYS

NEW DRIVER! ALLOWS WINDOWS 3.0 TO BE EXPANDED ACROSS FOUR MONITORS

CEBRA COMMUNICATIONS LIMITED

26 LORNE PARK ROAD, BOURNMOUTH
TEL 0202 299048 FAX 0202 299192

CIRCLE NO. 894

DATA COMPRESSION LIBRARIES™

PKWARE's® Data Compression Libraries™ allow software developers to add data compression technology to software applications. The application program controls all the input and output of data allowing data to be compressed or extracted to or from any device or area of memory.

- All Purpose Data Compression Algorithm Compresses Ascii or Binary Data Quickly with similar compression achieved by the popular PKZIP software, however the format used by the compression routine is completely generic and not specific to the PKZIP file format.
- Application Controlled I/O and memory allocation for extreme flexibility.
- Adjustable Dictionary Size allows software to be fine tuned for Maximum Size or Speed.
- Approximately 35K memory needed for Compression, 12K memory needed for Extraction.
- Compatible with most popular Languages: C, C+ +, Pascal, Assembly, Basic, Clipper, Etc.
- Works with any 80x86 family CPU in real or protected mode. $295.00
- No runtime royalties.

RUNNING OUT OF EXPENSIVE DISK SPACE?

PKZIP can help! PKZIP compresses your files to free up disk space and reduce modem transfer time. You can compress a single file or entire directory structures with a single command. Compressed files can be quickly returned to their normal size with PKUNZIP

Software developers can reduce the number of diskettes needed to distribute their product by using PKZIP. Call for Distribution License information.

The included PKZIP utility lets you store compressed files as a single self-extracting .EXE files that automatically uncompresses when run. Only $47.00

PKWARE® INC. THE DATA COMPRESSION EXPERTS®

9025 N. Deerwood Dr.
Brown Deer, WI 53223
(414) 354-8699
Fax (414) 354-8559

CIRCLE NO. 895

Digging Deep into Disks!

File-based disk access cannot be used for everything. To write sector editors, boot-virus scanners and disk compressors, you need a detailed knowledge of the disk's structure - like Simon Shepherd's.

The bottom-most view of a disk is simply a piece of rusty neoprene or ceramic. The disk controller chip has no idea whatever about files, it simply moves it's read/write head to a commanded position on the disk and transfers data when it is told to. Access to the disk at this level is via heads (which side of the disk), tracks or cylinders (concentric rings), and sectors (radial subdivisions of each track). It is up to you to locate the volume, track, sector and so on, and this in turn requires a detailed knowledge of the structure of the disk.

FAT Formats

MS-DOS allocates space to files in units called clusters. Each cluster can in theory contain any number of sectors as long as it is a power of 2 eg 1,2,4,8 etc. When a file needs more space, MS-DOS simply allocates it more clusters. The clusters allocated are stored in the file allocation table (FAT), and this critical structure is one of the most important data areas in any low-level disk manipulation program. MS-DOS can use two types of FAT: 12-bit or 16-bit. The FAT entry for each file forms a chain, each link pointing to the next cluster occupied by that file, until a terminating entry is reached. The first pointer into the FAT for the file is recorded in the directory entry for the file.

In earlier versions of MS-DOS, all FATs were 12-bit. This allows each FAT chain to point to cluster numbers as large as 4095 (less a few for other bits and pieces). Such a disk is limited to around 4080 clusters, which even at one sector per cluster works out to around 2 MB. At the time, this was considered vast, and never likely to be exceeded. For disks larger than 2 MB, we can cope simply by increasing the number of sectors per cluster. This was the initial approach for the 10 MB drive on the XT, where the allocation size was increased to eight sectors per cluster. The problem with this method, however, it that it is very wasteful. Even a 1 byte file occupies 4 KB of disk space.

The situation was remedied in Version 3 of MS-DOS with the introduction of the 16-bit FAT. We can now address up to over 65,000 clusters, and even one sector clusters cope with disks up to 32 MB. (It is worth pointing out, however, that each FAT would require 131,000 bytes of storage, or 512 sectors at 512 bytes per sector for both copies! To keep system overhead as low as possible, the MS-DOS designers limited the number of FAT entries to 16,384, thus keeping the 32 MB limit to four sectors per cluster and cutting the FAT down to 128 sectors).

Byte Offset	Field Length	Sample Value	Meaning
00h	word	0200	Bytes per sector
02h	byte	02	Sectors per cluster
03h	word	0001	Reserved sectors
05h	byte	02	Number of FATs
06h	word	0070	Max root dir entries
08h	word	02D0	Total sectors on disk
0Ah	byte	FD	Media descriptor
0Bh	word	0002	Sectors per FAT
0Dh	word	0009	Sectors per track
0Fh	word	0002	Number of heads
11h	double word	00000000	No of Hidden Sectors
15h	11 bytes	---	Reserved
*** Amendment for DOS 3 ***			
15h	double word	00000000	Total sectors if word at offset 08h = 0.

Figure 1 - The Structure of a DOS BPB

Anyway, I digress... We are interested in it being able to accurately determine all the low-level parameters for any disk, but most important, the FAT format of the disk.

The Boot Record

An important data structure which is present on every MS-DOS formatted disk, whether it is 'bootable' or not, is the BIOS Parameter Block (BPB) which tells MS-DOS all about the disk. It starts at offset 11 in the boot sector, which must *always* be the first sector on the disk or volume. Its structure is shown in Figure 1.

From the BPB it is possible to calculate almost everything else we need to know about the disk. Note that, given the usual sector size of 512 bytes, the largest disk that can be addresses is 32 MB. This is because the largest value the word variable `total_sectors` can have is 65535, which when multiplied by 512 bytes per sector gives the 32 MB limit. In version 3 and later, the BPB has been extended to allow for logical drives larger than 32 MB by the addition of an extra field (Figure 1).

From this, we can determine everything we need to know to address the disk at low level. For example, we know the hidden (or boot) sectors are always first on any disk and we know how many there are, so we know where the first FAT starts. We know how many FATs there are and what their sizes are. This tells us where the root directory starts, and since we know how many entries it can contain (and each entry is always 32 bytes), given the number of bytes per sector we know where the data area starts and so on. However, we are **not**

Byte Offset	Field Length	Sample Value	Meaning
00h	byte	80h	Boot indicator 00h = non-bootable 80h = bootable
01h	byte	01h	Starting head
02h	byte	01h	Starting sector (bits 0-5)
03h	byte	00h	Starting cylinder (plus bits 6-7 from above entry)
04h	byte	04h	System ID 00h = unknown 01h = DOS 12 bit FAT 04h = DOS 16 bit FAT 05h = DOS extended partition
05h	byte	04h	End head
06h	byte	51h (11)	End sector (bits 0-5)
07h	byte	0E9h (1E9h)	End cylinder (plus bits 6-7 from above entry)
08h	double word	00000011	First partition sector
0Ch	double word	0000A2A1	Sectors in partition

Figure 2 - The Structure of the Partition Table

explicitly told the format of the FAT. As we have said earlier, it is vitally important to get this right, because any mistakes will result in a disk full of spaghetti where there were once files!

Floppy FATs

Diskettes always have a 12-bit FAT. The 2 MB limit mentioned above for 12-bit FATs/one sector per cluster suffices for all formats of floppy disk, and yet all floppies adopt two sectors per cluster as their unit allocation size. This can waste up to half the space on the floppy, and half the space in the FAT! No-one has ever explained this bizarre anomaly.

Hard FATs are harder

But what about hard disks? The simplest approach is to say that small hard disks will have a 12-bit FAT and large hard disks will have a 16-bit FAT. We can even be a little more sophisticated, and try to work out from the BPB data what format of FAT the disk would most likely have, given the amount of storage space to be addressed and the space allocated to the FAT itself. This method is straightforward and often works, but there is always the exception which will trip you up - result: disk pasta!

One of the reasons we are going to look in detail at how the extended partitions work is the ability to determine accurately the FAT format MS-DOS has given to any particular logical drive, however it has been partitioned on a fixed disk.

Partitions

Hard disks may contain a number of logical volumes, and indeed, more than one operating system. It is perfectly possible to have MS-DOS on one part of a disk and XENIX on another part. To cope with this extra data, fixed disks have a more complex structure. The five basic important areas are:

1. The Master Boot Record containing the partition table .
2. A Boot Record containing the BPB for each partition.
3. A FAT (normally two copies) for each partition.
4. A Root directory for each partition.
5. Data space for each partition.

Almost every hard disk has a master boot record, located on the very first physical sector on the disk, addressed as Head 0, Track 0, Sector 1. (Don't forget that BIOS numbers heads and tracks from zero, but its sectors from 1. This is the exact opposite to MS-DOS, which numbers its logical sectors from zero.) This master record is loaded at boot time by the ROM BIOS, and is responsible for reading the partition table at the end of the sector and acting accordingly. If the partition table indicates that the hard disk contains a bootable partition, then the boot sector for the bootable drive is located and loaded, which in turn loads the rest of the operating system. If the hard disk does not contain a bootable partition, then the bootstrap code attempts to load an operating system from floppy. Failing that, true blue machines default to ROM BASIC.

The partition table describes how the hard disk is apportioned, and always maintains a rigid data format. Within the partition table there is sufficient data space to allocate as many as four partitions on the hard disk, each with its own entry. Notice the offset of the data table - the first entry starts at 01BEh, the second at 01DEh and so on. The final word AA55h is the usual MS-DOS validity identifier. The format of the data within the partition table is summarised in Figure 2.

There are several things to notice about the partition table. First, the boot indicator tells the system whether the partition is bootable. Only one of the four partitions can have this capability. The system ID describes which operating system is using the partition. Only a few sample values have been shown, but others are possible. UNIX, XENIX, Pick and so forth will have their own unique identifier bytes. The format of the partition table is fixed rigorously, not only across versions of MS-DOS, but across all PC operating systems. Note that the start

Offset	Size	Function
00h	byte	Request header length
01h	byte	Unit code
02h	byte	Command code
03h	word	Status word
05h	8 bytes	Reserved
0DH	byte	Media descriptor
0Eh	double word	Transfer address
12h	word	Sector count
14h	word	Starting sector number

Figure 3 - DOS Block Device Request Header for Read/Write

THE SOFTWARE CONSTRUCTION COMPANY

BORLAND

Borland C++	£199
Turbo C++ 2nd. Edition	£55
Turbo Debugger & Tools	£85
Turbo Pascal v.6.0	£69
Pascal Professional	£129
Pascal for Windows	£129
ObjectVision	£79
Paradox 3.5	£375
Paradox Engine	£215
Paradox SQL Link	£249

AUTHORISED

DATABASES

Advanced Revelation	£795
Advanced Revelation -	£695
askSam	£255
askSam LAN	£579
Clarion Personal Dev. 2.0	£79
Clarion Professional Dev.	£679
DataEase	£485
DataEase - LAN 3-PAK	£485
Dataflex	£449
Dataflex Multiuser	£1065
Knowledgeman/2	£645
Magic PC	£325
Open Access III	£449
Open Access III Compiler	£385
R:BASE 3.1	£515
R:BASE Compiler	£190
R:BASE 3.0 LAN Pack	£645

DBASE

Clipper 5.0	£299
dBASE III PLUS	£395
dBASE IV	£495
dBASE IV Developer's Ed.	£695
dBFast/DOS	£255
dBFast/PLUS	£255
dBXL	£165
dBXL LAN	£389
Force	£449
FoxBASE+	£295
FoxBASE+/386	£425
FoxPro	£515
Vulcan	£320

OBJECT-ORIENTATED

Smalltalk/V	£65
Smalltalk/V 286	£129
Smalltalk/V Windows	£325

CASE TOOLS

EasyCASE Plus	£190
EasyCASE Plus Prof.	£255
EasyCASE Prof. Upgrade	£80
Visible Analyst	£385
Visible Analyst Workbench	£2460
Visible Analyst Workbench	£6995
Visible Analyst Workbench	£1295
Visible Dictionary	£595
Visible Prototyper	£595
Visible Rules	£595

VERSION CONTROL

MKS RCS	£125
MKS RCS - 5 User	£485
PVCS Professional	£319
PVCS - 5 User	£1345
SMS	£395
Sourcerer's Apprentice	£175
Sourcerer's Apprentice	£429

TEXT AND UNIX TOOLS

MKS AWK	£65
MKS AWK DOS & OS/2	£129
MKS Programmer's Platform	£429
MKS Toolkit	£165
PolyAWK	£129
PolyAWK Toolkit	£145
SpellCode	£65

TRANSLATORS

BASTOC	£515
BAS_C Commercial	£579
dBx/dBPort	£389
dBx/dBPort with source	£645
FOR_C	£359
Pascal to C Translator	£99

Microsoft

BASIC 7.1	£219
C 6.0 Compiler	£210
C 6.0 & Windows SDK	£349
COBOL 4.0	£345
FORTRAN 5.0	£199
MASM 6.0	£59
Pascal Compiler	£149
QuickBASIC	£50
QuickC	£50
QuickC/Assembler	£85
QuickPascal	£50
Windows 3.0	£65
Windows S.D.K.	£260

Authorised

EDITORS

BRIEF for DOS & OS/2	£199
BRIEF upgrade	£60
BRIEF with dBRIEF	£275
BTags	£35
Cheetah	£159
dBRIEF for dBASE/Clipper	£99
dBRIEF for Paradox	£99
dBRIEF for R:BASE	£99
EDT+	£190
EMACS	£210
Epsilon	£129
KEDIT	£99
MKS Vi	£99
Multi Edit Professional	£115
Norton Editor	£50
SPF/PC	£159
SYNDIE	£319
VEDIT PLUS	£120

LINKERS

.RTLink	£190
.RTLink/Plus	£319
ALINK	£129
Blinker	£125
Plink w/LTO	£359

MAKE UTILITIES

MKS Make	£99
MKS Make - 5 User	£320
PolyMake	£129
PolyMake - 5 User	£485

PARSERS AND LINTS

MKS Lex & Yacc	£165
PC-Lint	£90

PROFILERS

.RTLink/Plus	£319
CHARGE	£65
Codesifter	£79
Inside!	£85
PC Metric	£129

DEMO BUILDERS

Dan Bricklin's Demo II	£190
Instant Replay III	£99
Instant Replay Prof.	£385
Show Partner F/X	£259

MEMORY MANAGERS

386 MAX Version 5	£85
Above Disk	£79
Blue Max	£100
HEADROOM	£85
HEADROOM Network Extens	£39
HI386 Complete	£65
MOVE'EM	£59
QEMM-386	£65
QEMM-50/60	£65
QRAM	£55
Turbo EMS	£65

DOCUMENTATORS

4c for Brief	£79
ABC Flowcharter	£190
ASMFLOW	£129
C-Clearly	£85
C-Doc	£120
Clear for C	£129
Clear for dBASE	£129
dAnalyst for C and C++	£190
dAnalyst for Clipper	£190
dAnalyst for dBASE III	£190
dAnalyst for dBASE IV	£190
dAnalyst for FoxBASE +	£190
EasyFlow	£99

FREE Microsoft Flight Simulator

*WORTH £45 WITH EVERY PRE-PAID ORDER OVER £100**

Flow Charting III	£165
SourceDoc	£190
The Documentor	£190
Tree Diagrammer	£65

DISASSEMBLERS

Sourcer 486	£85
Sourcer w/BIOS pre-proc.	£110

E-MAIL

cc:Mail	£450
cc:Mail EXPAND	£385
cc:Mail Remote	£190
DaVinci eMAIL	£645
DaVinci eMAIL Combo	£965

CODE GENERATORS

DATABOSS	£395
Matrix Layout	£195
ObjectVision	£259
ObjectVision for Windows	£259
PRO-C	£259
R&R Code Generator	£99
UI2 Developer's Release	£385
UI2 Touch & Go	£259
Zachary	£259

DEBUGGERS

Periscope 286/386 Pod	£389
Periscope I OK	£319
Periscope I/512K	£385
Periscope I/MC	£449
Periscope I/MC 512K	£449
Periscope II	£145
Periscope II-X	£129
Periscope IIw/Switch	£145
Periscope IV (16 MHz)	£775
Periscope IV (25 MHz)	£900
Sherlock	£129
SVS DBG	£210

MULTI-TASKING

DESQview 2.2	£85
DESQview 386	£125
Dr. Switch-OnCall	£65

BATCH ENHANCERS

Batcom	£39
Beyond.BAT	£65
Command Plus	£85
ProKey Plus	£85
The Builder	£99

TERMINAL EMULATORS

BLAST II	£195
COTERM/100	£145
COTERM/220	£160
COTERM/220	£160
COTERM/4010	£165
EM 320	£199
EM 4010	£230
EM 4105	£330
EM 4105 Network	£345
ICE.10	£190
SmarTerm 125	£190
SmarTerm 2392	£65
SmarTerm 240	£225
SmarTerm 320	£129
SmarTerm 340	£225
SmarTerm 400	£115
SmarTerm 4014	£145
Smarterm 470	£190
ZSTEM 100	£65
ZSTEM 220	£99
ZSTEM 240	£190
ZSTEM 4014	£65

BRIEF 3.1

- Mouse Support
- EMS Support
- Dual mode DOS & OS/2
- Dual media 3.5" & 5.25"

FREE Logitech Serial Mouse with BRIEF 3.1 still only £199*

*All prices exclude V.A.T. U.K. carriage included. One offer per order at published prices.

PAYMENT

VISA or ACCESS. Credit to approved accounts.

CATALOGUE

Call for your free copy of our 160 page catalogue, TOOLBOX.

1 THE MALTINGS, GREEN DRIFT, ROYSTON, HERTS. SG8 5DB
TELEPHONE (0763) 244114 FACSIMILE (0763) 244025

CIRCLE NO. 896

Partition Type	Usage	Approximate DOS Partition Size
01h	DOS 12 bit FAT	0 - 16 MBytes
02h	XENIX	
03h	XENIX	
04h	DOS 16 bit FAT	16 - 32 MBytes
05h	Extended Partition	
06h	DOS 3.31 16 bit FAT	32 - 512 MBytes
75h	PCIX	
DBh	CP/M	
FFh	XENIX bad block table	

Figure 4 - Summary of System ID Bytes

positions of the partitions are cumulative, a subtle but vital fact which most reference books omit.

The function of disk partitioning is to separate the disk into logically separate areas. Each is assigned a separate drive letter by MS-DOS, and behaves as though each were a physically separate drive. In addition, as we have seen, it is possible to partition a disk for more than one operating system.

Beyond text books

This is basically the state of play as summarised in nearly all the MS-DOS programmers' reference books, and ought to be very easy. A utility to handle any disk type could perform an analysis something along the lines of:

1. Is the disk floppy or fixed?

2. If floppy, read physical sector one to get boot record. Extract data from BPB about disk. Assume 12-bit FAT.

3. If fixed disk, read physical sector one to get the master boot record. Extract FAT format from system ID byte. Extract location of desired partition from partition table, read that physical sector to get logical drive boot record, and then extract BPB data as before.

Well, if you just have say a C: and D: on your hard disk, and both of them are smaller than 32 MB, then you might just get by with the above approach. But hard disks are getting bigger, and many machines now come with 110 MB as standard, with much larger drives still available off the shelf. This gives us the option of wanting many more logical drives than just the four allowed in the master partition table, so what do we do now?

The method adopted to overcome this limitation is referred to as extended partitioning, introduced with MS-DOS 3.3. Although it is an important aspect of disk operation, most technical references are conspicuously quiet about the subject.

By way of an example, let us look at a typical 40 MB hard disk on a PS/2 Model 70, partitioned under MS-DOS 3.3 as C: D: E: and F:. We expect that a cursory look at the partition table using Norton's Utility, Virus Hunter or some similar disk utility should reveal all four entries in the partition table are occupied with data on the four logical volumes. No way, Jose! You will find that only the first two have entries, the rest contain zeros. Where is the partition data for E: and F:?

The reason for this phenomenon is that every logical volume of the disk has its own extended partition table. The master partition and the extended partition tables form a chain pointing from one table to the next until the last partition is reached. This would clearly be necessary if we had partitioned the disk into 20 logical drives, since the master partition table would have insufficient entries, but note that the systems often adopt this strategy even when it isn't strictly necessary. The explanation seems to be that the system assumes that you may use FDISK at some time in the future to add more partitions, and is taking precautions against that day! The key to understanding the whole process is in the system ID byte, found at offset 04h in the partition table (see Figure 2).

By the way, the terminology is often used very loosely when talking about extended partitions, but it is important not to confuse the different terms. Under simple partitioning, the disk is divided into different MS-DOS partitions, each of which is treated as a logical volume and assigned a drive letter by MS-DOS, as we have said. Under the extended partitioning scheme, the disk is divided into a primary MS-DOS partition and an extended MS-DOS partition. The primary partition is treated as one logical volume and assigned one drive letter, but many logical volumes may reside within the extended partition, with each volume being assigned a unique drive designator.

In our example, the first table entry, corresponding to the C: drive is the primary MS-DOS partition. Examination of the first entry in the master partition table reveals that there is 04h in the system ID byte. We deduce from this that the C: drive is formatted for MS-DOS and has a 16-bit FAT format. We can read the start head, cylinder and sector directly from the table, and use BIOS interrupt 13h to read the boot sector associated with C: directly for us. From this, we can read out all the other data we require from the BPB, starting at offset 11 in the boot sector.

But what of D:? We see an entry of 05h for the system ID byte. Our trusty reference book infers that this means an extended partition with a 16-bit FAT format. Well, the first conclusion is right, but the second is dangerously misleading! The entry 05h does indeed indicate an extended partition, but *never* infer that the D: drive has a 16-bit FAT format. It may have, but don't rely on it.

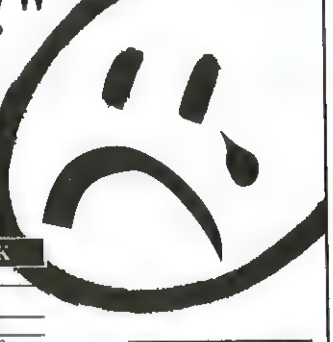
The correct FAT format for D: is found by reading the partition start data for D: from the master partition table, adding on the size of C: (don't forget the partition offsets are cumulative). Calling this head, track and sector through BIOS will take us directly to the boot record for D:, which is the first logical sector (MS-DOS sector 0) in the D: volume, the first volume in the extended partition. But where has MS-DOS put the D: extended partition table? The secret is to work back until you come to the first sector on that particular track, head 0. There you will find the extended partition table for that particular logical volume. We can now read the table. In my example, the first entry will be for the D: volume itself, which we find has a system ID byte of 01h. (D: has a 12-bit FAT after all!)

Partition Type	Size MB	Sectors (Clusters)	Sectors Per Cluster
1	0 - 16	< 4086	8
4	16 - 32	< 65536	4
6	32 - 64	< 131972	4
6	64 - 128	< 262144	8
6	128 - 512	< 1048576	16

Figure 5 - Default formats of various partition sizes

"AS YOU LIKE IT!"

Copy Protect AND/OR
Copy Register...Don't give it
away & cry!



EVERLOCK:
Stops copying & controls
programs as you want it.

EVERTRACK:
Does not copy protect
but it does control.

EVERLOCK
EVERTRAK

EVERLOCK

- COPY PROTECTS YOUR PRODUCT
- POSITIVELY SECURE AGAINST 'BIT -COPY' SOFTWARE
- PREVENTS DE-BUG, TRACING & DISASSEMBLY
- SUPPORTS ALL HARD & FLOPPY DISC FORMATS
- COMPATIBLE WITH MOST NETWORKS
- OPTIONAL RESTRICTION OF NETWORK USAGE
- LIMITS PROGRAM USE BY EITHER NUMBER OF DAYS AND / OR EXECUTIONS
- REMOTE RESET OF INSTALL, DATE, EXECUTIONS
- SUPPLY UP TO 99 INSTALLS ON ONE DISC
- NO DAMAGED MEDIA OR NON STANDARD 'DOS' ACCESS
- SUBSTITUTE I O PLUG FOR SOFTWARE
- END USER TRANSPARENT
- EASY PRODUCT UPGRADE
- "SECURE STRING"- EG USER REGISTRATION / SERIALISATION

EVERLOCK
NEW RELEASE
EVEN MORE
POWER

EVERTRAK

- NON COPY PROTECTION OF SOFTWARE
- AUTOMATIC PRODUCT SERIALISATION / REGISTRATION
- ANTI - REVERSE ENGINEERING
- OPTION TO ATTACH PASSWORD
- LIMIT NETWORK CONCURRENCY
- DYNAMIC PROGRAM COMPRESSION
- DYNAMIC PROGRAM ENCRYPTION
- END USER TRANSPARENT
- COMBINE EVERLOCK & EVERTRAK

EVERTRAK
NEW RELEASE
EVEN MORE
POWER

DUPLICATION SERVICE

ANY QUANTITIES LARGE OR SMALL

WITH OR WITHOUT PROTECTION

HIGH QUALITY STATE OF THE ART EQUIPMENT

TURNKEY PRODUCTION AVAILABLE

AS YOU WANT IT!

USER FRIENDLY

Tel 0527 510 105
Fax: 0527 514 229

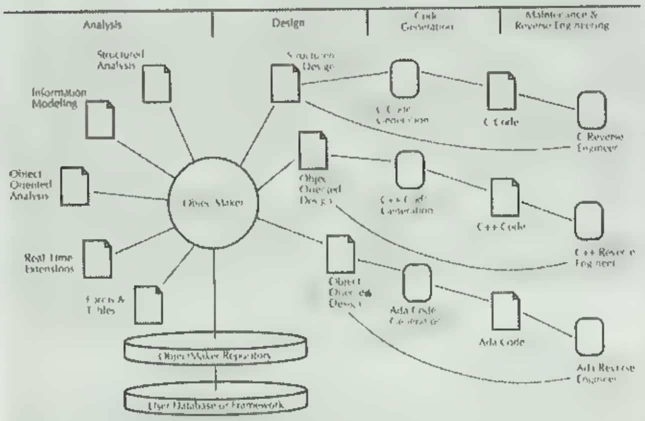
22A Bartleet Road
Washford Industrial Estate,
Redditch
Worcestershire
B98 0DG ENGLAND

CIRCLE NO. 897

ObjectMaker®

A trademark of Mark V Systems

The Customisable *CASE* Tool........



.......With Off The Shelf *Windows*™ Solutions.

Traditional & Object Oriented Methods

ADARTS
Bailin
Berard
Booch 86 & 90
Buhr 84 & 90
Coad/Yourdon

Colbert
Constantine
Chen
DeMarco
Firesmith
Gane/Sarson

Harel
Hatley/Pirbhai
Martin
Shlaer/Mellor
Ward/Mellor
Yourdon

Language Support
Ada C and C++

Windows
MS, X11, DEC, RISC, Sun Open & Open Desk Top

Operating Systems
MS-DOS, Unix, AIX & VMS

Available in Europe from **EUROLINE** Systems

Tel: (44) 926 55168

20 Foresters, Bicester Road, Oakley, Bucks HP18 9PY

All trademarks recognised

CIRCLE NO. 898

System Science

C COMPILERS

Microsoft C 6.0 with PWB	£245.00
Quick C & Quick Asm 2.51	£99.00
Quick C ver 2.5	£65.00
Borland C++	£239.00
Zortech C++ Dev Ed	£275.00
Watcom C 8.0	£255.00
Aztec C86 Comm	£265.00
C compilers for 386	£call

C GRAPHICS & DATAFILE

Halo	£195.00
GSS GDT	£445.00
Metawindows	£199.00
Essential Graphics (no roy)	£275.00
PCX Prog Toolkit (new 4.0)	£165.00

C DATA & MISC

Btrieve for DOS	£345.00
XQL for Btrieve	£475.00
Ctree Plus	£345.00
CodeBase 4.2 (source)	£175.00
CodeBase ++	£175.00
GL Financial Mathlib	£245.00
Blaise C Tools Plus	£89.00

C SCREENS & COMMS

Cscape, Look & Feel	£365.00
Vermont Views & Designer	£325.00
Panel Plus II	£275.00
C Worthy & cwARCHITECT	£call
Essential Comms (src)	£195.00
Blaise C Aynch Manager	£115.00
Greenleaf CommLib	£185.00

PROGRAMMERS TOOLS

SPE prog editor	£189.00
MKS Toolkit (Unix toolset)	£155.00
PVCS Prof	£375.00
PolyMake	£135.00
CProgrammer Toolbox	£159.00
Brief Editor 3.0	£195.00
Kedit Editor	£110.00
Flow Charting 3	£150.00
Dan Bricklin DEMO II (new)	£195.00
MS Progs Lib 1.2 CD	£225.00
Personal Rexx	£110.00

ASSEMBLERS & LINKERS

MS Macro ASM 5.1	£90.00
Turbo Debugger & ASM 2.0	£99.00
RTLink Plus	£345.00
Plink 86 Plus	£295.00
Sourcer & Bios (DisAsm)	£125.00

CROSS DEVELOPMENT

Introl C or Modula-2	£1445.00
Aztec C Cross	£275.00
IAR C Cross with CSpy	£call
2500AD Cross AsM	£135.00
Simulators & others	£call

FORTRAN

MS Fortran-77 5.0	£235.00
Lahey F77L-EM/32 & 386 OS	£845.00
Salford FTN77/386	£765.00
Lahey F77L	£375.00
RM Fortran Forte	£475.00

PASCAL, BASIC

Turbo Pascal (6.0)	£75.00
Object Professional (for TP)	£110.00
Turbo Pascal Windows	£125.00
Blaise Power Tools Plus	£89.00
Blaise Asynch Plus	£89.00
MS Basic 7.1 with PWB	£255.00
MS Quick Basic 4.5	£65.00
PowerBasic (ex Turbo Basic)	£85.00
many Pascal, Basic libraries & tools	£call

OTHER LANGUAGES

Smalltalk V/Windows	£295.00
TopSpeed Modula-2	£145.00
MS Cobol 4.0 (new ver)	£445.00
RM Cobol-85 ver 5.0	£995.00
LMI ur Forth	£225.00
PC Logo	£50.00
Clipper 5.0	£395.00

MEMORY, MATHS etc

386 Max 5.0	£75.00
QEMM/386 5.1	£60.00
QRAM for 286	£45.00
Above Disc (LIM Emul)	£69.00
Norton Utilities (new ver 5)	£110.00
Spin Rite II	£69.00
Derive (Symbol Alg) 2.0	£135.00
MicroStat II	£295.00
Mathematica by Wolfram	£call

WINDOWS 3.0

MS Windows 3.0	£85.00
MS Windows SDK & MS-C	£435.00
MS Windows SDK 3.0	£310.00
Case: W New Corp User	£CALL
WinPro/3 from Xian	£525.00
WindowsMaker	£595.00
Toolbook by Asymetrix	£295.00

"OUR RANGE INCLUDES MANY MORE ITEMS FOR DOS, OS/2, UNIX ETC. PLEASE ENQUIRE"

**3-5 Cynthia Street,
London N1 9LF
Fax: 071-837 6411**

Prices are subject to change and alteration Prices include UK Delivery but are exclusive of VAT.

071-833 1022

CIRCLE NO. 899

```
AL                          Drive Number (A=0, B=1 etc)
CX                          -1 (0FFFFh)
DS:BX                       Pointer to packet

Format of data packet pointed to by DS:BX

packet                      label        byte
Starting Sector Number      dd           ?
Number of Sectors           dw           ?
Transfer Address            dd           ?
```

Figure 7 - Interface to Ints 25h/26h from DOS 3.31

We can now deal with E: and F: by repeating the same procedure, starting by reading the second entry from the extended partition table just located. When we get to the last extended partition table for F:, we see that the second entry is all zeros, and we know we have finished. The only limit on the length of the chain of partition tables is the number of drive letters available to MS-DOS. Since A: and B: are reserved for the floppies, the maximum number of partitions is effectively 24, each being up to 32 MB in size, resulting in a maximum disk usage of about 768 MB.

Complications

Some manufacturers of top-end machines, such as Compaq, couldn't wait for poor old MS-DOS to catch up with their expectations, so they wrote their own versions of MS-DOS with amended disk drivers to cope with the monstrous hard disks they were fitting. Try reading the system areas of some Compaq 110 MB disks with most disk utilities - even the mighty Norton wheezes and gives up!

As we have seen, MS-DOS 3.3 introduced the ability to partition a disk into up to 24 volumes of up to 32 MB each. Compaq very reasonably argued, however, that it was ridiculous to have to partition a 300 MB disk into 10 logical volumes if you didn't want 10 drives on your machine! DOS Version 3.3 as published by Compaq (as MS-DOS 3.31) introduced a disk partitioning scheme that conforms to the MS-DOS limit of 24 volumes per drive, but dispenses with the 32 MB limit, allowing volumes as large as 512 MB each to be created. Using all 24 possible volumes, this scheme allows disks up to 12.25 GB to be partitioned.

But the mental arithmetic wizard reading this article will have spotted one small problem. A 512 MB drive, assigned to one volume, would have 1 million 512 byte sectors, which, even with eight sectors per cluster, would lead to a rather large FAT of some 1000 sectors. To add to the problems, various internal MS-DOS data structures, such as the variables in the request headers used for transferring data to and from disk device drivers, were unable to cope with sector numbers larger than the word limit. Accordingly, Compaq overcame the problem by stipulating that their very large partitions would have more sectors per cluster (as many as 16 sectors per cluster for partitions from 128 to 512 MB) and by rewriting the structure of the device drivers in the IBMBIO.COM system file.

Bigger Headers

The format of the request header for a conventional MS-DOS block device driver READ/WRITE command is shown in Figure 3.

It can be seen that another reason for the 32 MB limitation on volume sizes before MS-DOS 3.31 is the word variable describing the starting sector number for the disk address. What Compaq did was alter this variable to a double word to accommodate the increased sector addressing requirements. In all other versions of MS-DOS, the length of request headers is always 22 bytes, but device drivers rarely, if ever, check this quantity. They simply assume that MS-DOS knows what it is doing when it builds its data structures! Under MS-DOS 3.31, however, the request header length is now 24 bytes, and the disk drivers in IBMBIO.COM check this size to determine whether they are being passed requests for the special extended partitions. As a double check, the device attribute word, found at offset 04h in the driver itself, has bit one set to identify the extended capabilities of the driver. This bit was previously undefined for block devices.

So now all that is missing in the scheme of things is a system ID byte to allow Compaq to indicate that one of its huge partitions is in use. The identification byte 06h was assigned to this job, and always implies that a 16-bit FAT is in use. So the current state of play with ID bytes is summarised in Figure 4.

AL	Drive Number (A=0, B=1 etc)
CX	Number of sectors to read/write
DX	Logical start sector number
DS:BX	Data transfer address

Figure 6 - Interface to Ints 25h/26h up to DOS 3.3

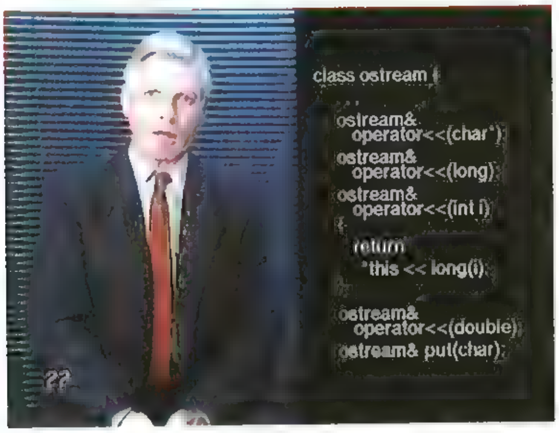
The default formats of the various partition sizes are summarised in Figure 5. Note that the type 1 partition maximum size is based on the number of clusters in the partition as opposed to the absolute number of sectors. Partitions with 0 - 4085 clusters are type one. Partitions with 4086 or more clusters (but less than 65536 total sectors) are type four. Confused?

Finally, a few amendments to the MS-DOS kernel itself were necessary. Interrupts 25h and 26h, which MS-DOS uses for reading and writing absolute disk sectors, required some modification to again overcome the limitations on word variables. The register interface for standard MS-DOS to these interrupts is summarised in Figure 6, while the amended interface is shown in Figure 7. It is interesting to note that when standard MS-DOS finally caught up in version 4, it adopted this format as well. One crucial difference exists, however. Under MS-DOS 3.31, it is still possible to use the old 16-bit sector call, and the interrupt will function correctly. This is very useful because existing software works correctly. In addition, the interrupts will protect data in the large partitions from being inadvertently changed by applications that do not understand the format of the extended addressing. Should an attempt be made to address a sector beyond the 32 MB limit without using the extended addressing, a 'Sector Not Found' error is simply returned.

MS-DOS 4, however, requires that the full extended interrupt format be adopted even if the sector being addresses lies within the 32 MB limit and could be addressed via the existing 16-bit format.

EXE

Simon Shepherd is Technical Director of Defiant Systems Limited and acts as an MS-DOS consultant on a number of software projects. He can be reached on 0752 265777.



```
class ostream {
  ostream&
    operator<<(char*);
  ostream&
    operator<<(long);
  ostream&
    operator<<(int i)
    {
      return
        *this << long(i);
    }
  ostream&
    operator<<(double);
  ostream& put(char);
```

That Software Tools Man

Peter Collinson has been visiting AT&T's famous Bell Labs to speak to the man who owns UNIX user id no. 9 - Brian Kernighan.

You're a Canadian, aren't you? How did you get from Canada to Bell Labs at Murray Hill in New Jersey?

I was an undergraduate at the University of Toronto, graduating in 1964. It looked fairly clear in 1963/1964 that to continue graduate education in Computing there was only one Canadian choice - staying at Toronto - but there were a number of plausible options in the United States.

After a fair amount of thinking, I decided to go to the United States, on the grounds that it didn't seem to be a good idea to stay in the same school for a very long period of time. I had already been at the University of Toronto for four years, and had lived most of my life in Toronto. So it was about time for a change. I picked Princeton for several reasons, none of which are exactly good reasons retrospectively but all of which were important at the time.

The main reason might have been that they offered me full financial support and promised to get me out in three years. At best, most other places would offer an Assistantship and the dim hope that I would get out in six or seven years. This seemed like a very clear-cut trade. The three years was a lie, but I did manage to do a PhD in four and a half years. This was better than the seven that MIT had promised. My degree was technically in Electrical Engineering. It was in the very early days of Computer Science at Princeton, and it was a sub-component of Electrical Engineering.

How did you end up at Bell Labs?

I came here straight away. I had spent one summer working at MIT in the very early days of Project MAC in 1966. I enjoyed that. At the time Bell Labs and General Electric were collaborating with MIT. I could see people who I had never met, like Ken Thompson, logging into the machines in Cambridge, where I was, from Bell Labs in Murray Hill, where I had never been. The following summer, the summer of 1967, I managed to get a job at the Labs; this was more or less by happenstance but also by the good offices of Ted Dolotta. I worked with Doug McIlroy, at least officially. I had a great time.

Brian Kernighan is probably best known as the 'K' in 'K&R', but his influence extends well beyond the famous White Book. Based at AT&T Bell Labs, he has participated in the development of UNIX from its birth pangs to the modern vogue for Open Systems. He is a great communicator and has co-authored several influential and readable books on software design and the UNIX philosophy. These include Software Tools with PJ Plauger and The UNIX Programming Environment with Rob Pike. For UNIX itself, he is responsible for much work on text processing including device independent troff. He also co-authored awk, a language designed for writing quick-and-easy text processing programs. Kernighan still works at AT&T Bell Labs.

Doug has had a very long standing interest in storage allocation because he is responsible for `malloc()`. My project, insofar as there was one, was to think about investigating various kinds of storage allocation algorithms. I got almost instantly sidetracked into the intricacies of list processing. I spent the summer building a very simple package of list manipulation routines that you could use from FORTRAN programs. It was reminiscent of something called `MADSLIP` which had been done at MIT by Joe Weizenbaum (who originally did Eliza). It let you do list processing in FORTRAN and was actually fairly convenient. I spent the whole summer on it, so I really didn't do a thing about storage allocation.

I came back the following summer and by then I was into thesis writing or thesis researching mode. I worked with Shen Lin on combinatorial optimisation problems, of which graph partitioning was the first - I did my thesis on that. The other main problem was the travelling salesman problem. I came back for real, early in 1969. I'd had so much fun here that I never looked anywhere else.

What happened to get you involved with the UNIX group?

At various times in 1967 people like Ken Thompson, Dennis Ritchie and Joe Ossanna were working on a MULTICS machine here at Murray Hill. Somewhere, probably in 1968, that MULTICS machine went away because it was very expensive. It looked like the MULTICS system wasn't coming together at any reasonable speed. It was just too big, too slow and too complicated. The machine was either pulled out or converted into something else.

I was in that same centre, the Computer Science Research Centre, so physically I was part of the group. However, although I did have friends on that corridor, I was not technically involved. Then at some point, presumably in 1969, Ken started to build a system on the PDP-11. Ken, Dennis and Rudd Canaday were right across the hall from me. I would see them arguing about the file system design. I remember when Ken had his system actually working. There was something that you could actually do some work with, and I chipped in the name UNIX, although not spelled the way it is today. I was thinking 'UNICS'. I do not know who spelled it as we have it now. I got in early enough that I have a single digit user id, 9, on a lineal descendant of the one true UNIX. There are very few people who can claim that distinction; it's basically Ken, Dennis, Doug and me.

Software Tools

One interesting strand of your work is the idea of 'tools'. How did you get involved with the whole 'Software Tools' thing?

> ***Ratfor didn't add much to FORTRAN, but it did convert it into a programming language***

Plauger and I wrote the *Elements of Programming Style*. That worked out pretty well and was fun. The idea behind the style book was to take a large number of programs and criticise them: that isn't right, that could have been better. We weren't saying how to do things, but how not to do them. It was all in FORTRAN and PL/1, which at the time were the dominant languages in the community we were aiming at.

A couple of years later, we decided that the time had come to tell people about how they ought to do things. By then, we had a clearer picture of some of the benefits of the UNIX environment: the advantage that you can get by piping programs together and building things that were going to be filters. What we wanted to do was to convey the good ideas of UNIX programs, tools, and approach people who didn't have UNIX available.

It wasn't clear what to use as a language for the book. C, of course, didn't exist in very many environments. I had already designed and implemented `ratfor` (Rational FORTRAN); it simply stole the good appearance of C. It didn't add much beyond that, but it converted FORTRAN into a programming language. So we decided to use `ratfor` as the programming language for the book. This was really unconventional.

FORTRAN was one of the two really portable languages at the time, so it was curious that you didn't do the book in it.

Of course, our contention was that `ratfor` was infinitely easier to read than FORTRAN. I still believe that. At the same time, it produced FORTRAN so it could be portable.

The original version was written in C with a small amount of `yacc` grammar. Given that as a bootstrap, it took a very short time to write it in `ratfor`. It could then be bootstrapped on a machine only running FORTRAN. The book came with a tape of all the code and part of the tape was `ratfor` in `ratfor`. The first thing on the tape was `ratfor` in FORTRAN so you could just peel it off and start running with the code.

So you have `ratfor`, how do you get from there to Software Tools?

Then you sit down and ask what are the things that are interesting in a UNIX system. It's the fact that there are a large number of very small tools - `cat`, `wc`, `cmp` and so on - that you can glue together in interesting ways. Sorting is one of those tools as well; even though it's not really a filter, packaging `sort` as a filter makes it more useful and teaches a useful lesson. That took us through the first four chapters of the book: the ideas of filters; input/output redirection; character input and output as the common denominator; text is everything; and data is just streams of lines of text.

Then we did regular expressions. These are a very important fundamental notion in UNIX. Using this, the next couple of things

Ratfor

Ratfor stands for **Rat**ional **FOR**TRAN. The basic concept is to have a pre-processor that reads some arbitrary syntax and generates FORTRAN. The FORTRAN can be compiled by a native compiler and run. In fact, the language is not *that* arbitrary - it allows you to express many of the constructs of C while really using FORTRAN. Also from necessity, there are many bits and pieces of FORTRAN in the language - you don't want to have to do too much work.

Ratfor is very akin to C in the sense that it has all of the familiar control structures: `if`, `while`, `for`, `do` becomes `repeat. It` replaces many of the standard `FORTRAN` conditional tests with their C equivalent so `.EQ.` becomes `==`. Blocks in Ratfor are shown using the familiar curly brackets, however, functions and routines map directly onto FORTRAN subroutines. If you are a C programmer and you get hold of an original copy of *Software Tools* by Kernighan & Plauger then you will have no difficulty in reading the Ratfor and understanding it.

For example, here is a complete Ratfor program that reads and writes single characters:

```
# copy input to output
        integer getc
        integer c

        while (getc(c) ¬= EOF)
                call putc(c)

        stop
        end
```

The ¬ is the *not* symbol.

What Prolog programmers have been waiting for !

Prolog-2 for Windows
version 3.0

Once again Expert Systems Ltd. are first to deliver a Prolog system matched to today's programmer's needs.

Our new Prolog-2 for Windows 3 is the only package that allows programmers to combine the power of Prolog logic programming with the acclaimed GUI of Windows 3.

Now **you** can create 32-bit applications for the world's favourite GUI with **your** favourite programming language.

Just look what Prolog-2 for Windows 3 gives you :

- Full 32-bit addressing using 386 enhanced mode, no more limits on workspace, heap etc.
- Virtual memory support
- Startling speed. Up to 150 kLIP on a Research Machines VX486
- Full bidirectional C interface for embeddability
- Integrated window editor and graphics
- Over 500k of Hypertext help
- Workstation performance on a PC for just £1495.00

I need a Prolog to write Windows 3 applications right now. Please rush me Prolog-2 for Windows 3.

Please add £10 p&p plus V.A.T.

My cheque is enclosed
or My official order is attached
or I wish to pay by Mastercard/Visa
All trademarks acknowledged

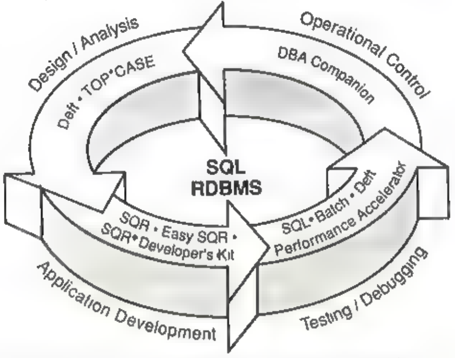
copies at £1495 (£747.50 acads)
Card no.
Expiry Date
Name and address
Signature

Expert Systems Ltd.

Unit 12, 7 West Way, Oxford, OX2 0JB. England. Tel 0865 794474 fax 0865 250270

CIRCLE NO. 904

SQL Solutions "THE" Systems Integrators



SQL Solutions is the first single source vendor of professional services and SQL productivity tools to provide complete SQL Systems Integration.

We offer:

- ✶ **RDBMS independent Productivity tools to address each phase of the applications development lifecycle.**
- ✶ **Unparalleled expertise in the design and implementation of applications in all major RDBMS's.**
- ✶ **Gateway products to enable realtional databases to co-exist.**

To see our full range of products visit us on stand 17a at the Software Tools exhibition

SQL Solutions (UK) Ltd
Kennet House, Waterside Park
Sweetwell Road
Bracknell, Berks, RG12 1HH
Tel: 0344 714053 Fax: 0344 714056

CIRCLE NO. 906

C-Compilers The American Option

Developed with 12 years experience, true processor specific ANSI and K&R C-Compilers are now available complete with free Assembler and Linker.

1802, 1805, 6301, 64180, 6502, 6800, 6801, 6802, 6805, 6809, 6811, 68000, 8051, 8085, 8086, 8096 and Z80.

ONLY £595

and now 86C010 only £895

Development System support for over 150 devices is also available, complete with High Level Debugging, to see you through your complete software and hardware design cycle.

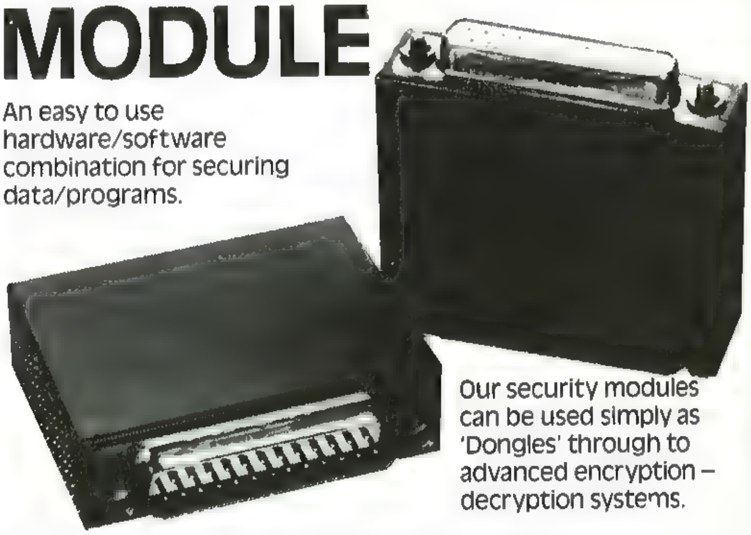
Call us now for your **FREE** active demonstration disk

american automation
uk oxford Tel: (0993) 778991 Fax: (0993) 702284
Lea View House, Two Rivers Est, Station Lane, Witney, Oxon OX8 6BH

CIRCLE NO. 905

SOFTWARE SECURITY MODULE

An easy to use hardware/software combination for securing data/programs.



Our security modules can be used simply as 'Dongles' through to advanced encryption – decryption systems.

- ★ For IBM PC family (and compatibles)
- ★ Variants for other systems
- ★ Advanced construction – rugged and compact
- ★ Proven throughout the world
- ★ Transparent operation – user friendly
- ★ Driver software supplied for variety of applications

CTL
Control Telemetry of London
11 Canfield Place, London NW6 3BT
Tel: 071-328 1155 Fax: 071-328 9149

CIRCLE NO. 907

are pattern searching: the `grep` family, and then the editor.

Well, what's the rest of it? The only other thing that we do in UNIX is text formatting, so we wrote a simple formatter. Finally, we did a macro processor, and then `ratfor` itself, written in `ratfor`.

Was all the work involved making all these programs done as an internal exercise, or was it done for the book?

It was all done for the book. The idea for the book came first and the code was developed for that. `Ratfor` was independent of the book but it became the vehicle for the book. The FORTRAN version of `ratfor` was intended for portability and in particular, for the book. The version that was described in the book was simpler than the version that was on the tape because you didn't learn anything new by having the full implementation described to you.

You could hold in your hot little hand some code that would run on your VMS machine or whatever. People picked up all the code and used it for real.

Yes, that's what happened. There were groups like the Software Tools group that sprang up at the Lawrence Berkeley Lab in California, with Debbie Scherrer, Joe Sventek and Dennis Hall. They set up the Software Tools group and they did some really nice work with the tools code to create the 'Virtual Operating System', a portable UNIX-like system that would run on a number of machines. It was all done in a very clean way.

There is still some `ratfor` around, because I occasionally see companies saying that they will translate `ratfor` to C. This is a trivial extension of their ability to translate FORTRAN into C. I periodically get calls from somebody who has inherited a very large `ratfor` program and wants to know what to do with it.

Awk

Can you tell me a bit about awk?

`Awk` dates from 1977. It's by far the biggest software project that I have ever been involved with. There were three of us involved, and that's completely unworkable. Somehow, it's much easier working with two rather than three because it's harder to split things up among three people. There's more divergence of opinion. Sometimes that's good because it means that there are more good ideas but sometimes it's not as cohesive as it might be. On the other hand it was very nice to work with Al Aho and Peter Weinberger.

I have been working on a program to convert awk to C++

How did the ideas for awk arise?

Three separate threads came together. I had been interested in programmable editors but kept thinking that there had to be a better way of manipulating text. I wanted something that would allow me to manipulate both text and numbers.

Then I came across a tool that was done by Marc Rochkind, who was at the Labs at that time. This was basically a data validation tool. It took pairs of regular expressions and messages. It scanned through an input file looking for something that matched one of the regular expressions; when something was found it printed the corresponding message. In some sense, that's the basic notion of `awk`. There is a bunch of patterns and something to be done with each one of them.

Al Aho was the world expert on regular expressions. Peter Weinberger had a strong background in databases. `Awk` is in no sense a database tool but it has some of that flavour, and that comes from Peter's database experience. So we took this mishmash of ideas and glued them together.

Around 99% of my awk usage are those tiny applications like column printing and summing. Yet some people have written huge scripts in it.

It was meant originally for writing these one and two line programs. It really was. I think it's very seductive because it does so many things automatically. It handles strings and numbers smoothly. It is an interpreter and there's no baggage, no object files. People start to write a one or two line program that just grows and grows; some of them grow unbelievably large: tens of thousands of lines - which is nonsense.

Was awk something that has continuously evolved? Was the second generation of awk an evolution or a decision to change things dramatically?

There was always a slow background set of changes to `awk`. At one point I returned to my notion that I really wanted a programming language where I could write programs that would manipulate text as easily as most programming languages can manipulate numbers. I thought that rather than going in the direction that C had gone from B, I would

Awk

`Awk` is a program on UNIX used to process text and numbers. It is a *filter* in the UNIX sense. It takes in a stream of bytes, performs some operation on them and writes the result out. A simple `awk` program consists of a *pattern* and an action. Each input line is scanned for the pattern, and if the line matches then the action is taken. The simplest pattern is null, so every line is matched. A very common `awk` program uses this to print a single column from a set of input data, so

```
ls -l | awk '{print $4}'
```

takes the output from `ls`, splits the source data into columns separated by white space and prints column four on every line. Selecting a particular column is often useful.

`Awk` can also perform numerical actions on the data. A program to sum the numbers in a particular column and print the total is simple:

```
# /bin/sh
ls -l | awk '{sum = sum + $4}
       END  {printf "Total %d\n", sum}'
```

This demonstrates a more complicated program consisting of several pattern/action pairs. The first action adds the value of the number in column four to the running total held in sum. The last action is a magic one and is matched at the end of processing the input data. This triggers a conventional-looking printf statement to print the results.

`Awk` provides a very rich C based programming environment. Is is also approachable by non-programmers. I have often found users who have no formal programming experience will accept and use `awk` for day-to-day data processing tasks.

MULTIPLE INTELLIGENT COM PORTS FOR OS/2

THE IMPAC 8 & 16 INTELLIGENT COMMS CARDS

- ✻ MICROSOFT COM X COMPATIBLE
- ✻ 8 TO 64 FULL MODEM PORTS
- ✻ REDUCE CPU OVERHEAD FROM 40% TO 1% PER PORT
- ✻ FITS ALL ISA (PC/XT/AT) MACHINES
- ✻ DOES NOT USE SHARED RAM
- ✻ PROVEN FIELD PERFORMANCE
- ✻ UK DESIGN AND MANUFACTURE
- ✻ DRIVERS FOR DOS, OS/2, UNIX & XENIX

DEALER ENQUIRIES WELCOME

For more information contact:
SCL Downswood, Carlton Road
Reigate, Surrey RH2 0JQ
Telephone: 0737-762200

CDOS DRIVER ANNOUNCED

CIRCLE NO. 908

F77L-EM/32

Port 4GB mainframe applications to 80386s with this 32-bit DOS-Extender compiler. The winner of PC Magazines's 1988 Technical Excellence Award just got better. New Version 3.0 and OS include: Editor, Make Utility, Virtual Memory Support, DESQview Support, New Documentation and Free Unlimited Runtime Licenses. F77L-EM/32 and OS/386. POA.

F77L

The compiler of choice among reviewers and professionals. Includes a Debugger, Editor, Profiler, Linker, Make Utility, Weitek and 386 Real-Mode Support, Graphics. POA.

Lahey Personal Fortran 77

New Version 3.0: Full ANSI 77, Debugger, Editor, Linker, Library Manager, Microsoft and Borland C interfaces, 400 page Manual. POA.

When people talk about FORTRAN
the name mentioned most often is
Lahey

System Science

3-5 Cynthia Street,
London N1 9JF
Tel: (071) 833 1022
Fax: 081 837 6411

CIRCLE NO. 909

DOS – UNIX "QICPAK–ONE"

1/4" DATA CARTRIDGE INTERCHANGE DOS-UNIX/-DOS

- Time/Date stamping of files preserved
- On the fly end of line conversion
- Full mapping of DOS/UNIX file names with optional creation of unique names under DOS
- Handles device and linked name entries
- Compatible with Apollo, Mentor Graphics, Sun, SCO UNIX etc.
- Full programmes interface.
- Source code of VTAR, Vutil, Vtrans supplied in Microsoft C.
- 200 page reference manual, controller card and ribbon cable.
- Software only and complete solutions available.
- Optional facilities to process and duplicate cartridge from other systems.
- Support is by the developers.

VOGON
ENTERPRISES LIMITED
94 Easthampstead Road,
Wokingham, Berkshire RG11 2JD
TEL: 44 (0)734 784511/890042
FAX: 44 (0)734 890040

CIRCLE NO. 910

RM /Cobol-85 RRP £1050.00

- Ver 5 adds features incl. pop-up windows
- GSA certified ANSI Cobol-85 compiler
- Compiles to portable object code.
- Available for DOS, OS/2, AIX, Unix, Xenix.

RM /Panels RRP £330.00

- Forms design and Screen Managment system
- WYSIWYG Rapid screen prototyping

RM /CO* RRP £170.00

- Application development environment
- Edit, compile debug and project management.

Special Offer

RM Cobol-85, RM/Panels and RM/CO* for DOS at our very special price of £995.00 plus VAT, only from System Science

Prices are exclusive of VAT, but include UK delivery

System Science
3-5 Cynthia St, London N1-9JF
Tel: (071) 833-1022 Fax: (071) 837-6411

CIRCLE NO. 911

go in the other direction. I invented a language that I called A, a very simple thing that took the fundamental notions of awk and wrapped them up in a programming language. It didn't work very well. I played around with it but it was never very good. The implementation was really flaky, so I never used it much and nobody else ever used it.

Then it occurred to me that I could take the ideas and put them into awk. Usually, awk processes data by applying pattern matching to each input line but if you wanted to *do something* and not process data then the program could be placed in an awk BEGIN block. You simply wouldn't have anything else in the script. There was no need to have a separate language; I could just make it part of awk.

So I retrofitted things like the function mechanism, which is the new capability that really distinguishes it from the old version. Then later on, we made the regular expression matching dynamic and added the substitution commands. All the rest is trivial.

When you wrote the book did you look at what you had and say 'Gosh this is inconsistent and hard to explain; shall we change it?'

We did change it a small amount. I regret that we didn't do it more, because awk is quite irregular. There are lots of things that are stupidly designed and pointlessly illogical. There are things that have been left out. There are all kind of things that would have been much better if we had smoothed them out more than we did.

In part, we didn't do that because the version we were running had in some form been out for quite a while - at least a couple of years. System V release 3.1 had a version of awk that had functions. We were afraid that we would be unable to get a newer version out very easily so we felt constrained not to change it very much. In spite of that, we went and changed things, probably more than we should have and probably not the right things.

At the moment there are a couple of other implementations of awk around. In particular there is the GNU version called gawk. It's quite close to our implementation. There was a lot of contact back and forth while they were writing that to try to make sure that the versions stayed in sync as much as possible. There are also versions for MS-DOS. The MKS version is the one I know best; again that's quite close to what we have. Finally, there's a POSIX specification. POSIX is standardising awk; that's in draft stage at this point.

Is that crystallising what's in the book?

Well, more or less. Mostly I think that it is. There are some fine points to be resolved. In the draft standard there are some things that conflict with the book. POSIX takes one view and we take another on at least a couple of points. For the most part I think everybody agrees on the mainstream stuff.

Is the most recent version of awk going out with System V, release 4 as a standard?

My belief is that the version that comes out with SVR4 is quite current. Essentially, it does not differ from what we have except in a small number of bug fixes. The version that you can get from the AT&T Toolchest is closer to what we run because it can track more easily. I can just send them a new version. So both of those are within a tiny distance of what I run here.

Software

Someone's got to

- *write it*
- *design it*
- *use it*
- *approve it*
- *manage it*
- *pay for it*
- *justify it to the Board*

The Event for Software

- *Writers*
- *Designers*
- *Users*
- *Approvers*
- *Managers*
- *Purchasers*
- *Justifiers*

Software Tools Exhibition & Conference
11-13 June 1991 Wembley Exhibition & Conference Centre, London

Special features at this year's event include:

- A **Strategic Conference** with special business orientated sessions
- A **PC Development Area** Supported by **Microsoft**
- Free **Technical Training Courses** from Select Software Tools
- A **Business Solutions Clinic** for free expert consultancy
- A **People and Skills Centre** – addressing today's human resource issues

Make this year's visit to Software Tools a team event. It makes good business sense.

□ I am interested in exhibiting at Software Tools '91

□ I am interested in visiting Software Tools '91

□ I would like information on the Conference

For full details plus your show planner and free tickets just return this coupon to Lynne Davey, Blenheim Online, Ash Hill Drive, Pinner, Middlesex HA5 2AE, Before 3rd May 1991.

H

Name

Initials Mr/Ms

Company

Job Title

Address

Telephone

Fax

.EXE 6/91

CIRCLE NO. 914

SOFTWARE TOOLS

Organised by

BLENHEIM ONLINE

In association with

.EXE Software MANAGEMENT

Supported by

Blenheim Online
Blenheim House, Ash Hill Drive, Pinner,
Middlesex HA5 2AE, UK.
Telephone: 081-868 4466 Fax: 081-868 9933
Telex: 923498 ONLINE G.

G.W. Computers Inc
ADCI, 4 Eagle Square
E.Boston, MA02128
U.S.A.

Tel 617-569-5990
Fax 617-567-2981

Xvale Ltd., Store 'D'
55, Bedford Court Mansions
Bedford Avenue, London
WC1B 3AD U.K.

Tel: 071-636-8210
Fax 071-255-1038

Menu-IX | System-IX | DBMS-IX | Sysix | AI | Spell | Merge | Lock | GURU | Crypt | Tawk | Tgrep

We sell Hundreds of brand name DOS and UNIX software, and offer wide in depth experience on networks. CALL US!!.

LINX SYSTEMS

STANDARD FEATURES

1 * 5-1/4 " drive
1 * 3-1/2 " drive
Dual HD/FD Controller
101 Key Keyboard
Mini - tower case
2 Serial/1 Parallel/1Game
Built in Diag / AMI bios
Mono Screen / Adapter

CPU/ Mhz+Ram = Price	
8088 /10+640k	£545
80286/12+2mb	£745
80286/16+2mb	£825
80386/16+4mb	£1145
80386/20+4mb	£1345
80386/25+4mb	£1495
80386/33+4mb	£1745
80486/25+8mb	£2695

** For SVGA Colour**
Add Screen/Card £245.00
Then just add the
HARD-DISK of your choice

** LAPTOP portable **
80286-12,vga,40mb £2150.00
** NOTEBOOK portable **
80386-16,vga,20mb £1950.00

Unix 4 port	serial	£145
Unix 8 port	serial	£195
Intelligent 4	serial	£395
Genius mouse 3but		£45

Hard-Disks	
20 --	£150.00
40 --	£195.00
80 --	£325.00
100 -	£495.00
120 -	£595.00
200 -	£755.00
300-	£1095.00
600-	£1495.00
Ethernet	
wd8bit	£135
wd16bit	£155
lnx8bit	£110
lnx16bit	£135

INTERACTIVE R 2.2 UNIX PLATFORM		
Application	2 usr	£495
Application	3+usr	£1095
Network	2 usr	£695
Network	3+usr	£1350
Workstation	2 usr	£855
Workstation	3+usr	£1495
UNIX DEVELOPER		
Application	2usr	£1095
Application	3+usr	£1750
Network	2usr	£1155
Network	3+usr	£1815
Workstation	2usr	£1295
Workstation	3+usr	£1945

SCO	
Xenix o/sys	£422
Xenix s/dev	£455
Xenix tx/prc	£126
Xenix cmplt	£905
Xenix tcp/ip	£227
Xenix net	£451
Unix o/sys	£525
Unix s/dev	£613
Unix cgl	£191
Unix m/vw	£321
Unix vpix	£321
Unix tcp/ip	£227
Unix nfs	£340
Open dsktp	£795
O/dtp s/dev	£1095
O/dtp servr	£1095
Word 5.0	£650

HOT ITEMS	
Guru reprter	£49.95
Syseditor	£49.95
Cshell	£49.95
Menuix	£49.95
Sysix	£99.00
Sysix Pro	£145.00
SystemIX	£295.00
PC-Connect	£450.00
Pc-interface	£170.00
PC-TCP	£375.00
PC-Xview	£245.00
Terminl ctl	£695.00
Xvision	£295.00
Coherent ux	£99.00
DosBridge	£295.00

*** BUNDLED SYSTEMS include software Value up to £1730 FREE ***
Buy a 386/33 mhz with SVGA screen and 118 mbyte h-disk
at £2535.00 , get * INTERACTIVE UNIX Sys V, Rel 3.2, V2.02 *
RUNTIME / VPIX and SOFTWARE DEVELOPMENT + MENUIX
OR RUNTIME / TCP-IP and XWINDOWS + MENUIX
FREE !. CALL for latest deals. (24hr FAX)

** UNBEATABLE SYSTEM BUNDLES includes *FREE SOFTWARE* **

LINX 8088 - 10	LINX 80286 - 12	LINX 80386sx-16	LINX 80386-33	LINX 80486-25
mono/20mb	mono/40mb	colour/80mb	colour/120mb	colour/200mb
£695.00	£1020.00	£1760.00	£2535.00	£3695.00
£270.00 FREE	*£370.00 FREE*	*£800.00 FREE*	*£1300.00 FREE*	*£1730.00*

CALL FOR DETAILED SPECIFICATIONS
and any special offers

Limited time
offer on any
item of
software
IN HOUSE
while stocks
last !!.
Basic deals
are intended
to be 386
system
UNIX SYSV,
REL3.2,Ver2.02
Workstations
CALL NOW
and reserve
yours !.

Cut | More | Term Ctl | Where | Sort | Grep | Idle | C8 Trees | Evaluate | Awk | Banner | C Source | Fgrep | Help | Head | Tail

Xenix | Xenix Net | Unix | Interactive | System V | TCP/IP | DOS | Token-Ring | OS/2 | Ethernet | Novell Lan | Mic Net | Aix | VP-IX | VMS | QMS

Compaq | Toshiba | IBM | Epson | Qume | NEC | Seagate | Wyse | Relysis | Zenith | Priam | Ranger

CIRCLE NO. 915

The problem is that, for instance, a Sun user now will currently find that they are running a much older version of awk that doesn't match the book.

In many systems, there is an `awk` that is the old original thing and there is an `nawk` that is the new version. How I wish that somebody long ago had had the courage to name the older version `oawk` and call the newer one `awk`, rather than perpetuating this strange situation. That will get straightened out eventually but it's been a source of confusion for a long time.

There is a number of compilers around too. Having spent a lot of my life writing awk and shell scripts, one of my criticisms of the approach is that it doesn't scale up. At some point you have to throw the scripts away and resort to C.

There is an `awkcc` that was done here at the Labs by Chris Ramming and available through AT&T Toolchest. It is a nice piece of work, it really does pretty near guarantee to take an `awk` program and produce a C program with the same semantics. The C program will *do* the same thing and almost always run faster than the interpreter, probably on average twice as fast, and often even faster than that. The C program itself is utterly unintelligible, but it was never meant to be read.

What I have been working on, as a sort of back burner project, for the last year or more, is a program that will convert `awk` into C++. It will do it in such a way that the C++ is as close as possible to the original `awk` input. This is something that you can imagine doing in C++ because you can define a data type that captures the semantics of the `awk` variables and you can overload the operators so that you can write expressions that look like they looked in the `awk` program.

If this job is done well enough, when an `awk` program has gotten too big or needs some function that isn't part of `awk`, you can run the program through the translator. This creates a C++ program that looks so much like the original that you can just continue your development from there in C++. Alternately, you can be writing a C++ program and say 'I need an associative array, so let me just include `awk.h`' and be off and running.

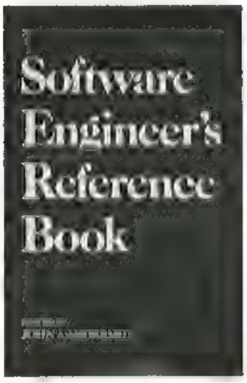
At the moment, it's sort of 95% done. I can handle all of `awk`; almost all of my regression test goes through. I had hoped that it would produce code that would run twice as fast as `awk`. I might even be able to do better, because I spend a lot of time doing type inference, saying 'that thing is only really used as an integer, let me declare it as an `int` rather than something more complicated'. Unfortunately, so far it is not a lot faster than the interpreter, which is sort of embarrassing. But the good news is that there is room for improvement! The output is very readable, which was my main goal, and I expect I can tune it to run much faster.

Many thanks to Brian Kernighan for generously taking the time to give this interview. Peter Collinson is a freelance consultant specialising in UNIX. He can be reached as `pc@hillside.co.uk` *electronically (although your mailer might be happier to put the address the other way round) or by phone on 0227 761824.*

Books

Software engineering - at a price - this month.

Interdisciplinary loot



Software developers are dirty thieves. We've been nicking ideas from other disciplines for years. We've 'borrowed' set theory, semantics and genetic algorithms. We hang around psychologists to find out about human-computer interaction. We half-inch stationery, pens, and bins to use in our desktop metaphors. Anything useful is dropped into programming. It's jealousy, of course; we pick out ideas from other disciplines because programming has no discipline of its own. That said, we're not as bad as we were. Out-and-out kleptomania seems to be coming to an end; as domesticated software engineers, we're beginning to plunder selectively and fixing the booty into a groundwork, just as mechanics and engineering has with physics and maths. We have a smaller set of targets. We don't nick now. We have underpinnings.

The arrival of books like this show how far we've come. *The Software Engineer's Reference Book* draws many different professions together to provide what its editor, John McDermid, hopes is an exhaustive list of what a software engineer should know. That such a book works as well as this demonstrates that computer programmers are beginning to get a feel for the limits and roots of the craft - I doubt whether a project like this was possible a few years ago.

That said, there's still rather a lot of sources to jam into the one book. The book is over 1000 pages long, with 63 chapters from as many academics and consultants. John McDermid, in his introduction, compares the design and implementation of the book to a large software project itself. Helpfully for critics, he provides a spec for the books coverage. It shows his objectives and the intended presentation to be carefully thought out : topics chosen were 'non-volatile' and mature; articles were to be written for practitioners rather than theoreticians. As each author writes to a separate brief, the choice of titles (or 'module specifications', as McDermid's continues the metaphor) was vitally important. McDermid seems to have got them on the nose: section headings pick out the most appealing spoils of the disciplines covered. There's unavoidable but minimal overlap. The authors were asked to provide overviews, but with sufficient detail to allow readers to get a feel for the extent and limitations of each area, and with a bibliography extensive enough to pursue in greater depth.

That, then, is the specification: but as every budding software engineer knows, there's many a slip between that and the ship. A lot depends on the development team. The list of contributors shows up McDermid's locale. Few authors hark from outside the UK ; eight share his campus at York University. A bit old boys network, but not, in fact, as bad as it seems: McDermid's choices are all competent professionals with enough grounding and writing skills to enlighten, and the evident coöperation between them provides the book with a greater fluidity than these compilations usually provide. Moreover, I have a sneaky suspicion that using big names with attitudes would encourage soapboxing, and detract from the neutrality and sedate manner of the tome. Plus (to be uncharacteristically selfish and partisan for a moment) the UK bias of the choice makes a welcome change. The British Telecom system used as the telephony application example has a familiar ring (ahem) to it, and the British and European standards institutions are given much better coverage than the skimpy dismissals I've seen in American textbooks. It's nice to see the bias on this side of the pond for once.

A latter day Von Neumann might be able to provide critiques of each and every chapter: I'm afraid as a monomath I can only dole out descriptions. McDermid divides the book into three sections; their full titles are 'Theory and Mathematics', 'Methods, Techniques and Technology', 'Principles of Applications'. The 14 chapters in 'Theory and Mathematics' provided the most captivating reads for me, but then I'm your classic graduate of Type-It-And-See Polytechnic. Still, even programmers whose middle names are APL symbols need to brush up on the latest advances. And I found Andrew Monk's chapter on the psychology of confusion handy, too. Lots of useful info on how to make a user feel stupid.

Much of 'Methods, Techniques and Technology' is more suited to DP managers than nose-to-the-metal programmers; on the other hand, the coverage within it of non-conventional programming languages is excellent. The mysteriously titled 'Principles of Applications' is a set of broad case studies : areas covered include computer-aided-manufacture, compiler technology, networks, databases, fault-tolerance and real time systems.

I think McDermid and his contributors can be proud of their work. It's annoying that the book is so very costly (a whopping 125 quid) and in only one volume. A three volume set with cheaper prices might have provided for a wider audience. As it is, it's all or nothing, and despite sterling efforts by McDermid to be omniscient, people can fall between the gaps in his coverage. Graphic systems programmers, for example, might, on forking out £125, be somewhat miffed by the peripheral coverage of their domain. On the other hand, the breadth of coverage does serve a useful purpose; the overviews of topics such as functional languages and cognitive psychology are areas that you wouldn't choose given a Pick 'n' Mix set of volumes. Everything smacks of eventual utility; I can't say you'll get your money back, but you will certainly leave this book a better programmer (software engineer) than when you went in. Buy it on the company budget.

Title: Software Engineer's Reference Book *Pages: 1025*
Editor: John A. McDermid *Publisher: Butterworth/Heinemann*
Price: £125 *ISBN: 0-7506-1040-9*

Books Received This Month

Software Engineering, 3rd Ed. by Ian Somerville	Addison-Wesley	£21.95	ISBN:0-471-52901-X	pp653
Microsoft BASIC V7.1 by Peter G. Aitken	Wiley	£26.50	ISBN: 0-471-52901-X	pp512
Writing Compilers and Interpreters by Ronald Mak	Wiley	£23.95	ISBN: 0-471-50968-X	pp512
RPG II/III 2nd Edition by Nancy Stern et Al	Wiley	£33.95	ISBN: 0-471-521965	pp700
Computer Mediated Communications by Matthew Rapaport	Wiley	£28.30	ISBN: 0-471-516422	pp370
Microsoft WordBASIC Primer by Russell Borland	Microsoft	£19.95	ISBN: 1-55615341-4	pp331

GIS...The Rising Star of IT?

Geographical Information Systems, better known as GIS were spawned during the 1970s from a 1960s CAD technology that enabled paper maps to be scanned and digitised into a computer format. The progress of GIS was restrained only by the prohibitive cost of computer hardware and the poor spatial handling capabilities of then available software. Since the mid 1980s, the fruits of research into spatial data handling combined with inexpensive PCs and workstations have made GIS viable and within the reach of a mass market.

In 1990, the world-wide revenues from sales of GIS was $612m. The figure for 1991 is predicted to be close to $1bn. despite the recession!

A GIS is any system that provides spatially-referenced information from a digitised map base. Tools within the GIS enable the data to be interrogated, manipulated, analysed and modelled interactively by the user. Through the skill of a software specialist, it is possible to construct extremely powerful graphically-represented decision support systems.

GIS is a truly international technology. Over 50% of the market is shared by US companies Intergraph, IBM, McDonnell Douglas, Prime and ESRI, the latter being the author of Arc/Info, one of the best known products. Despite this apparent US dominance, the UK fares well with Alper Systems, GMAP, Laser-Scan and Smallworld all enjoying international success with UK authored GIS software.

The relative youth of the GIS industry means that no ready-made pool of GIS Software specialists exists that the systems vendors and end-users can tap. As a consequence,companies import and cross-train skills from the software engineering market. That means good news if you can think spatially and are experienced in UNIX, 'C' programming and databases.

Alan Carnell, Concurrent Appointments

ANALYSTS AND 'C' PROGRAMMERS

£16K - £28K **M4 CORRIDOR/EUROPE**

European Subsidiary of a highly successful US based manufacturer offers the chance to enjoy benefits arising from growth exceeding 100%. Several new openings have now been created both here and Europe for analysts and programmers with skills in particularly 'C' together with one of the following:-

UNIX, MS-DOS, SQL, etc.

Already with a customer base reading like a 'who's who' of the pharmaceutical world, my client is about to bow to increasing demand for its products from new sectors of commerce and industry.

If you have any of the above skills and would like to know more about these ground floor opportunities call **Keith Wallis** now on **081 339 0099** days, or even at home evenings/ weekends on **0372 452401**. Alternatively you can post your C.V. to the address below or fax it to **081 339 0144**.

ALLTRONICS PEOPLE

Alltronics People, The Corn Store,
Manor Farm, 124 Manor Road North,
Thames Ditton, Surrey KT7 0BH

LONDON (CITY)
ORACLE/UNIX
ALL LEVELS - To £24K + CAR

Programmers, Designers, Analysts and a Project Leader required for major expansion.

THROUGHOUT THE U.K.
IBM 6150 PERSONNEL
ALL LEVELS - To £30K + CAR

This appears to be the growth area presently, Whatever level you are at, wherever you are based we probably are aware of positions that will interest you.

SOUTH WEST LONDON
PC SUPPORT STAFF - To £19K

Windows, Graphics, Lotus 1-2-3, Novell. Top company. New developments.

GUILDFORD

C Developers & Snrs sought. Training in UNIX available for DOS people.
£12K To £19K Basic

NORTH KENT
ANALYST PROGRAMMERS
£15K To £24K

Financial Systems. C/SQL. Superb benefits.

MANCHESTER
ALL LEVELS - To £20K + CAR

C++, Multi-Tasking, Windows, Graphics. New area for long established Software Group.

W.MIDS/WARWKS./WORCS./STAFFS.
ELECTRONIC ENGINEERS
£13K To £20K

Hardware biased with some "Know-how" in either C or Assembler. This market is showing signs of life again!

W.MIDS./WARWKS./WORCS./STAFFS
AS/400 PERSONNEL
ALL LEVELS - To £30K + CAR

Seeking an improved position in this arena? Contact our offices now.

NORTH/B'HAM STAFFS
SOFTWARE SUPPORT - £15K + CAR

UNIX/Accounting Software.

CAMBRIDGE

A)	HW/SW ENGRS.(Comms)	To £24K
B)	SW ENGRS & SNRS "C"	To £22K
C)	C Programmer/Analysts	To £19K
D)	CAD/CAM Software Developers	To £22K

NORTH STAFFS

A)	68000 ASSEMBLER/C. Industrial Interfaces	To £17K
B)	Instrumentation Software Dev	To £16K
C)	VMS or OS/9 Dev.	To £18K
D)	PLC Programmers	To £17K

SOUTH BUCKS

Analyst Programmers & Snrs. and a Project Leader. Business Systems, UNIX/MS-DOS, ORACLE/C/SQL, NOVELL/WAN
To £25K Basic

ABACUS CAREERS L·T·D

3rd Floor, Smithfield House, Digbeth, Birmingham B5 6BS. Tel: 021-666 7484. Fax: 021-666 7430

FILE MANAGER

Software Tools Developers

Southern Herts **to £25K**

With unparalleled expansion plans, potentially the largest new commercial project in the U.K. and multi-national conglomerate financial backing, this company will rely heavily on the tools provided by its software development team. As part of this highly motivated team, you will be responsible for designing and developing advanced software tools using the latest relational and object orientated techniques. As a degree qualified software engineer with 2+ years C, C++, UNIX and ORACLE (or equivalent) experience this organisation offers career advancement second to none.

Future Systems

Berks **to £25K**

In order to progress your career from Software Engineer to Team Leader and beyond you will need to demonstrate a sound technical knowledge of C, UNIX, OOD, RDBMS, and X-WINDOWS/MOTIF. You will be given the opportunity to improve your own skills, whilst being responsible for the development of advanced Network Managment Systems. You will be assured exposure to emerging technologies, in order to determine their applicability to the systems you are developing and will be a vital technical member of a team dedicated to Network Managment within this repeated commercial communications company.

GUI Designers

Europe/London
to £32K + benefits

A unique opportunity to export your specialist GUI Design and Development skills to Europe. You will be presented with a working environment dedicated to exploiting leading edge technologies and promoting personal skills development. You will be responsible for designing complex applications and software systems, which will requiro an in-depth practical knowledge of one or more GUI standard: If you are degree qualified and can offer 3+ years experience of C, C++, UNIX, OS/2, PRESENTATION MANAGER, X-WINDOWS, MOTIF development in a formal environment then you could be playing a major part in the 1992 challenge of OPEN Europe.

C++, Windows

London **all levels to £30K + Car**

A superb opportunity to enhance and develop your creative Software skills on innovative products and projects. A chance to make real use of your existing skills as well as acquiring new ones on technically demanding commercial Software Systems within a leading results orientated IT company. With a good degree, experience of C, C++, Windows, OS/2, ORACLE, TOKEN RING LAN, UNIX and Structured Software development, the sky's the limit.

For more details

Contact Mike Jenkins for details on the above and many other vacancies on **0442 231691** days or 0442 253300 eves/wkends. Alternatively write to him at:

Executive Recruitment Services,
Hempstead House, Selden Hill,
Hemel Hempstead. HP2 4LT
or fax CV to 0442 230063.

* * WINDOWS DEVELOPMENT SPECIALISTS * *

Formed in 1985, QA Training Limited is now an acknowledged leader in the field of technical training with an international client list including many of the major corporates, software development houses and computer manufacturers. We are committed to quality and excellence in everything we do - and our reputation has been built firmly on the high calibre of our training staff.

We are seeking to further expand out team of Training Consultants, specifically in the field of WINDOWS. The role will offer you an unique blend of teaching, learning and consultancy, ensuring that you remain at the forefront of technological advances.

Courses and seminars with which you would be involved will vary depending on your experience, but would include

- Microsoft Windows Programming
- Advanced Windows
- Troubleshooting Windows
- Designing Windows Applications.

If you can demonstrate specific technical skills in the following areas

- **Programming in WINDOWS - extensive experience**
- **C programming - a minimum of 2 years experience**
- **Exposure to design methodology**
- **Knowledge of Presentation Manager and/or OS/2**
- **Real world development experience**
- **Ideally Project management experience**

then please send your Curriculum Vitae, quoting your salary requirements to the address below.

We would also be interested in hearing from candidates with specific and in depth experience in the areas of:

- UNIX/AIX - (C programming under UNIX, excellent knowledge of UNIX at the shell interface level, working knowledge of TCP/IP, NFS, BNU, X Windows, OSF Motif)
- NETWORK SYSTEMS - (Lans, Comms and Connectivity)

Melinda Bishop - Personnel Manager
QA Training Limited, Cecily Hill Castle,
Cirencester, Glos GL7 2EF

No Agencies Please

QA TRAINING

OPPORTUNITIES IN GIS SOFTWARE DEVELOPMENT

To £20,000 — gmap MODELLING & PLANNING GEOGRAPHICAL — **Leeds**

If you are a young software developer with experience of GIS or with a flair for creating complex database applications, then you'll be in excellent company if you join GMAP.

GMAP custom-built Spatial Analysis and Modelling Systems that provide the senior decision makers in top Private and Public sector businesses with decision support systems many had imagined were impossible to produce.

Central to our exceptional success is our varied client base both in Europe and USA, combined with the seemingly infinite number of application areas identified as lending themselves to our techniques.

To help us sustain our growth, GMAP wish to recruit a number of bright, young problem solvers with the ability to think out problems in a spatially-referenced context. You should be a person who enjoys finding your own answers rather than coding someone else's. You will need to be fluent in 'C', preferably under DOS and/or UNIX and be quick to learn other languages and operating systems, as required. Experience of graphics and database tools would be useful but not essential as training in these areas can be arranged.

Interested? To learn more about working with **GMAP** please call **ALAN CARNELL**, our retained recruitment specialist at Concurrent Appointments for an informal discussion or mail him your CV in confidence.

INTERVIEWS WILL BE HELD LOCALLY

Concurrent Appointments
Software Recruitment

27 FIELD CLOSE
HARPENDEN
HERTS
Tel: 0582 712976
Fax: 0582 764858

WEST YORKSHIRE

Programmers (particularly COBOL and 4GLs), Analyst Programmers, Software Engineers, Project Leaders, Support (pre and post sales), Trainers (MS-DOS, UNIX and Applications). Also Hardware/Software Design Engineers.

FOR YOUR NEXT CAREER MOVE AROUND WEST YORKSHIRE

Telephone Vincent Atherton on Leeds (0532) 504560 or write to:

AIREDALE RECRUITMENT

Realtex House, Micklefield Lane,
Rawdon, Leeds LS19 6AX

AIREDALE RECRUITMENT

INTERNATIONAL VACANCIES

GERMANY/SWEDEN £24K + CAR

European subsidiary of US software house involved in Image and Document Processing require software engineers with C or C++ on MS-DOS/UNIX using WINDOWS. Prepared to crosstrain from C to C++. English speaking environment, but knowledge of European languages useful for this pan European opportunity.

BELGIUM/ITALY ORACLE £25K

2 years' experience of SQL in a C/UNIX environment. English speaking environment.

ITALY SYSTEMS MANAGER £25K

Experience of TCP/IP, X25/OSI gained ideally in SUN/UNIX systems administration, knowledge of ORACLE and X-WINDOWS advantageous but not essential.

ITALY SENIOR DESIGNERS TO £35K

Extensive knowledge of UNIX real-time design together with team leading skills are required for this key role within an international software house. Spoken Italian or French useful but not essential.

ITALY C PROGRAMMERS £20K

One to two years' experience of C in an MS-DOS/WINDOWS environment.

CANADA CONTRACT I.E.F. 6/8m

USA CONTRACTS

CLIPPER V.5.0 7/8M ST LOUIS
DOS SUPPORT SPECIALISTS 7/8M

UK VACANCIES

ORACLE/INGRES/SYBASE
UNIX ENVIRONMENT, SURREY TO £25K

INGRES I.E.F. CONSULTANT
SOUTH LONDON TO £30K + CAR

INGRES/ADA/C/SOFTWARE ENGINEERS
CASE TOOL DEVELOPMENT, SURREY £20K

ADA SOFTWARE ENGINEERS
E/W DEVELOPMENT SURREY £22K

PAL PROGRAMMER BERKS. £18K

JUNIOR PC SOFTWARE SALES
SURREY £18K OTE + QUALITY CAR

CLIPPER DESIGNER/CONSULTANT
C. LONDON TO £25K + CAR.

THIS IS A BRIEF SELECTION FROM OUR CURRENT VACANCIES. IF YOU REQUIRE INDIVIDUAL CAREER ADVICE WITH DISCRETION CONTACT:

CHURCHILL RECRUITMENT CONSULTANTS

**The Old Bakery,
Commercial Way, Woking,
Surrey, GU21 1HW.
Telephone 0483 740441
Facsimile 0483 740466**

.EXE

An Announcement

During the next 6 weeks, .EXE Magazine will be conducting a readership survey.

The purpose of the survey is to monitor trends in the computer industry and help us make .EXE as useful as possible to you in your job.

The survey will be mailed out, with a pre-paid return envelope, to a balanced sample of our readership.

If you receive a copy, we would ask you to help us by filling it in and returning it to us as soon as possible.

Many thanks, in anticipation, for your help.

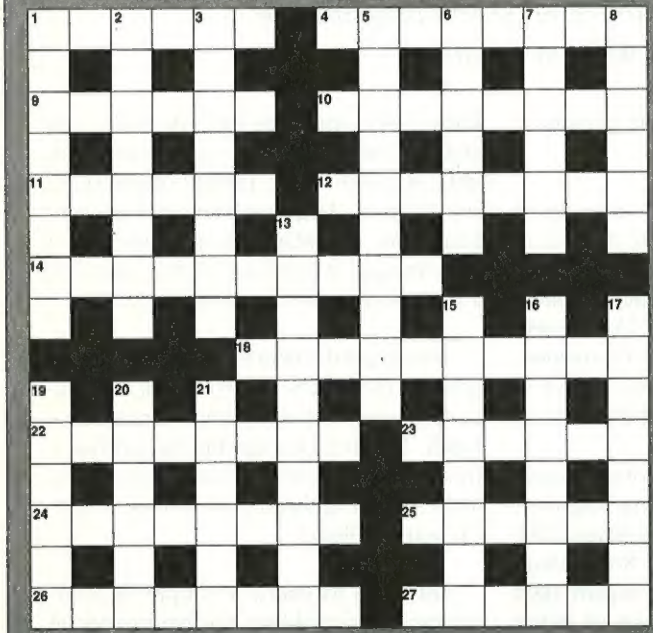
The Publisher, .EXE Magazine

Editorial Index

If you are interested in the News articles on pages 4, 6, 8 & 10, and would like to receive more information about the products mentioned, please circle the correspondong circle numbers on the Reader Service Card

AET	Coherent UNIX	920
Applied Logic	Conference proceedings	921
Avel	UPSs	922
Blaise Computing	Turbo Vision add-on	923
Borland	C++ compilers	924
Borland	Paradox Engine	925
Channel Business Systems	Disk Catalogue	926
ChyDale	Clipper OOP extensions	927
EQ Consultants	$25 Network	928
Fox Software	FoxPro	929
George Consultants	STEM library	930
Glockenspiel	Commonview	931
IDC	TSR & interface libraries	932
Innovative Data Solutions	PARAGen Code generator	933
Intek	C++ compiler	934
Intellicorp	KAPPA-PC	935
Micro Focus	COBOL add-ons	936
Microsoft US	Windows Resource Kit	937
MINIX Centre	MINIX OS	938
Nantucket	Clipper	939
NEOW	ToolBook	940
Noble Campion	Cameo image	941
Opensoft	Memory Commander	942
Panelhigh	M/4 for Windows	943
Phar Lap	DOS Extenders	944
QBS	Blinker	945
Real Time Associates	Modula-2 compiler	946
Rogue Wave	C++ class libraries	947
Scientific Software Ltd	GKS library	948
Software Construction Co	Codebase++	949
Solution Systems	C-Worthy	950
SSC	C reference	951
TEGL	GUI libraries	952
Tektite	Graphics card	953
Template Graphics Software	PHIGS library	954
UKIC	Internet consortium	955
WSA Consultancy	STD Code database	956
Zinc	C++ interface library	957

JUNE .EXEWORD



ACROSS

1 Use red hat to carry beads (6)
4 Background work on the tape deck? (8)
9 Guides to word processors are on top (6)
10 Send data across in Fortran's mitigation (8)
11 Somehow sortin' the first blocks (6)
12 Fascinate by use of ENTER (8)
14 Tiny mark of remark (10)
18 Getting things going on their own (10)
22 Rufus round a central part of the drive (8)
23 Pale sounding chunk of data (6)
24 Exciting source of system power (8)
25 I leave strange Martian with a code for success (6)
26 Using 22 fast to input data? (8)
27 Even without other keys can lose data (6)

DOWN

1 Station at the point of death (8)
2 Family link in modem db (8)
3 Interfaces for travellers? (8)
5 State above 1dn node (10)
6 End users rather new in gates (6)
7 In short, I am menu driven to be virus proof (6)
8 Place for yellow press between the columns (6)
13 Putting on accent on carrier (10)
15 Quite forbidden for one dead to use grid (8)
16 Large floppy strangely (8)
17 Start into the mysteries (8)
19 Specifications for underware I hear (6)
20 In 2D scanner is in secret (6)
21 Smallest unit of data in a fibre (6)

V	I	R	T	U	A	L		M	A	C	H	I	N	E
O		E		G		A		I		A		N		N
L	A	B	E	L		N	U	M	E	R	A	T	O	R
U		O		I		D		I		R		O		A
M	O	U	S	E	R	S		C	H	I	N	W	A	G
E		N		S		C				E				E
S	A	D	A	T		A	M	P	E	R	S	A	N	D
		E				P		A				T		
S	I	D	E	S	T	E	P	S		W	A	T	C	H
I				Y				S		A		R		I
L	I	A	I	S	E	S		W	H	I	L	I	N	G
I		I		T		I		O		T		B		H
C	O	L	L	E	C	T	O	R		I	N	U	R	E
O		E		M		E		D		N		T		S
N	U	D	I	S	T	S		S	E	G	M	E	N	T

MAY .EXEWORD

'.EXEWORD' compiled by Eric Deeson

ADVERTISERS INDEX

ADVERTISER	PRODUCT/SERVICE	CIRCLE	PAGE
Addison Wesley	Book Publishers	917	97
Al International	Quintus Prolog	864	38
American Automation	ANSI C Cross Compilers	905	89
Applied Logic Computing	Object Oriented Training	866	41
Arden Microsystems	Quick Basic Toolbox	871	45
Bits Per Second I	Graphics for dBASE	865	41
Bits Per Second II	Graphics Server Software	887	71
BL Computer Security	Software Protection	850	18
Borland	DOS and Window Programming	844	7
Brent Communications	MAX copy PROtection	913	92
Camel	Plotter Utilities	912	92
CEBRA Communications	Multi VGA Adapters	894	78
Clearsoft	Software Protection	862	35
Cocking & Drury	Smalltalk V	846	26
Comsec	Software Copy Protection	860	33
Creative Data Systems	Software Development Tools	902	86
CTL	Copy Protection Hardware	907	89
DES	Software Protection	903	86
Digital Equipment Company	Workstation applications	847	13
Euroline	CASE Tools	898	83
Expert Systems	Prolog for Windows	904	89
Genus	Flex OS & QNX	916	95
Great Western Instruments	C++ Embedded Software Design	867	41
Grey Matter	Programming Tools	842	3
GWI/Select	CASE Tools	849	18
HS Systems	8086 Emulator	883	62
IBM	CASE Solution	845	9
Instrumatic	C++	861	33
Intasoft	Software Management System	890	74
JPI	C++ Compilers	848/852	17/19
Koraia	Directory Utility	859	33
LBMS	Multi-user CASE	858	31
Linx	All UNIX Systems	915	94
LPA	AI/KBS/OOPS Software	882	62
Magnifeye	Software Protection Device	868	43
Microcosm	Copy Protection Software	901	86
Microft Technology	Security Software	886	68
Microsoft	Windows Development Software	855/856	25/27
NAG	Fortran 90 Compiler	840	OBC
Nu-Mega	Debugging Tools	873	47
Pinna Electronics	Tools & Graphics	851	19
PK Ware Inc	Utilities	895	78
Programmer's Odyssey	UNIX Software	869	44
Prospero	DOS Extender Kit	872	46
QA I	Windows Training	841	IFC
QAII	Software Testing.	876	53
QAIII	C++ GUI Class Library	880	61
Rainbow Technologies	Software Data Security	843	5
Rational Systems	DOS Extenders	853	21
Real-Time Software	CASE for Windows	877	55
Recital	RDBMS/4GL for VAX & UNIX	885	67
Roundhill	Development Tools	884	65
Salford Software Marketing	Fortran for DOS & UNIX	888	71
Second Computer Limited	Communications Boards	908	91
Sequiter	C Library	874	49
Signal	GUI Development Tools	870	45
Softlok International	Piracy Protection	889	72
S/ware Construction Company	Development Tools	879	59
S/ware Generation	Program Editor	878	56
Solution Systems	Programming Editor	896	81
SQL Solutions	Systems Integration	506	89
System C	Application Generator	891	76
System Science I	Development Tools	899	83
System Science II/Lahey	FORTRAN Compilers	909	91
System Science III	RM COBOL 85 Compiler	911	91
System Star	dbms Four C	857	29
The Data Business	Software Protection System	893	78
Unipalm	GUI Development	881	62
USA Software	Programming Tools	863	37
User Friendly	Software Copy Protection	897	83
Verilog	CASE Tool	875	50
Vleermuis	C++ Class Tree OS/2	854	23
Vogon	Data Interchange Software	910	91
Zortech I	Multi Plat C++ Comps	892	77
Zortech II	C++ Video Tutorial	900	85

On the job with Job

Too long on the Help Desk can transform the most patient programmer into a psychotic patient, claims William Campbell.

'Hello, you reported a problem. Yes you did. The Help Desk received a call from Violet half an hour ago. Couldn't she wait, or call tomorrow? Do you know what the problem is? No, don't look for someone!' (It will be all evening while he searches from sub-basement to attic for someone. An entire concerto composed of crackles, whirrs and faint voices plays from the ear piece. The 17:29 leaves Victoria Station.)

'Hello Dawn. What is "not working"? The thingamy. What bit of the computer thingamy is that? Is it the Branch Report? Are you sure?' (Mistake. Offer rope and Users will knot you with it. Disregard the seeded reply.)

'Did Violet leave a note?' (The telephone returns to intermittent Morse. The desk is always miles from the phone.)

'The Adjustment Report. What is "not right"? Can you print one off? Well that's something, being shown once.' (Warning - a little knowledge is worse than ignorance. This will take a long time.)

'Your niece's friend has a computer. A brown one. Yes, they are. The number of atoms in the Universe; that's really useful. So you have the cursor, the small blinking dash, next to the line saying "Adjustment Report"? Oh dear.' (Too much information. Users say what they think you expect to hear. Right line, wrong side of the screen.)

'Where is the cursor? What is immediately to the left? What did the error message say? You have to press the big key shaped like a backward "L", nothing else. No printing? Is the printer ready? The little square light marked "Ready".' (No chance of being ready for the 18:05. A rising panic at the call travelling backwards.)

'You waggled what cable! All the lights went out. Let's start from the beginning. Is the power lead still connected? I didn't say "plug" because there are several types. I know only one goes into the wall.' (*Ab initio ad nauseam*. Force calm pragmatism into tone and words. Instead of computer say thingamy. Jargon and terminology confound. An advantage for a Consultant, a disadvantage when fault tracking through a User. It means being multilingual.)

'You've got the little light on. No this isn't going to take much longer.' (Click. Realisation sprung from your empty crazed stomach. Two releases ago this Report had a freak miscalculation. Guess which ostriches never updated their system. A typical non-problem.)

'Just need to check a couple of fields against an old problem. You have to get on with the offices? Tonight is your floor polisher night.'

(On the Help Desk, the simple elements give you the mule kick in the fundamentals. It keeps your feet on the ground; and your hands and your spluttering breath.)

Verity Stob is unwell.

QUESTOR
Computer Personnel

APPLICATIONS DEVELOPMENT

Milton Keynes **£14 - £20k**

With at least 2yrs' development experience, you will be an enthusiastic programmer who is interested in developing quality applications software utilising sophisticated development tools. Good 'C' skills essential, MS-DOS & Windows 3, OS/2 & Presentation Manager are desirable. *Contact PH*

CAD SOFTWARE ENGINEERS

S. Herts **£16k - £33k + Car**

Highly regarded company established some 16 years, requires several additional applications development engineers to work on a 3D graphics product with colour imaging & near real-time walk throughs. Good C skills are essential, UNIX, Graphics or Fortran experience is beneficial. *Contact AH*

CLIPPER PROGRAMMER

E. Herts **to £17k**

This specialist software house designs and develops software for a range of major bluc-chip companies, primarily in the financial sector. We are retained to recruit several programmers with experience of writing SUBSTANTIAL applications programs in Clipper and dBASE. *Contact PH*

PROJECT MANAGER - dBASE

N. Bucks **to £20k**

Respected VAR developing database software to interact with graphics products requires manager to lead young small team. Good analysis, planning and leadership skills are essential for this key role. *Contact PH*

PROJECT MANAGER

Herts **£29k**

Technical Systems House are seeking a talented professional to take total accountability for the technical and commercial aspects of control system projects to around £2m. *Contact AH*

ORACLE DEVELOPMENT

N. Bucks **c. £17k**

2 years' experience of low level Oracle and embedded SQL on any platform, for state of the art R&D environment. *Contact PH*

TECHNICAL AUTHOR

Milton Keynes **to £20k**

Leading edge hardware/software company seeks a technical author with several years' experience of writing, preferably with an engineering bias. Experience of projects from concept to completion would be most useful. *Contact AH*

SOFTWARE ENGINEERS - VMS, UNIX, 'C'

S. Cambs **to £21k + bens**

Min. 2 years' exp' in a numerate degree subject, you will have knowledge of Fortran and C under UNIX, VMS or SunOS, coupled with CAD/CAM/CIM experience. *Contact AH*

SOFTWARE ENGINEER - 'C' & FORTRAN

S. Cambs **to £19k + bens**

R&D company has an urgent requirement for development staff for 2D and related products on VAX platforms. There will also be specification and design work involved. *Contact AH*

SOFTWARE ENGINEER - 'C' & FORTRAN

S. Cambs **to £19k + bens**

With 2+ years' esp' you will be interested in the development of Knowledge-Based manufacturing systems. VAX and Sun platforms. *Contact AH*

4GL DEVELOPERS

N. Herts **£ dep. on experience**

With a relevant HND or degree qualification, you will be self-motivated and enthusiastic with an interest in 4GL development. Sound career development potential with this established software house. *Contact PH*

Plus many other positions throughout London, South East England and the Home Counties. For further details of these or any other positions, send your CV in confidence or call Alan Hewson (AH, or Polly Hunter (PH) on:

Tel: 0462 438373
Fax: 0462 421272
Regal House
55 Bancroft
Hitchin
SG5 1LL

Evenings:

Alan Hewson **0462 459338**

QCP

They say the first Fortran 90 Compiler won't be available until at least 1992.

We say they're wrong.

NAG is pleased to announce the release of the worlds first Fortran 90 Compiler — one of several new exciting products to be released in the NAGWare Service.

Fortran is still one of the most widely used scientific and engineering programming languages and the f90 Compiler will ensure it stays so.

The NAGWare f90 Compiler is a full implementation of the ISO Fortran 90 Standard and is ready to meet the most demanding test of all — you.

For further infor … *following:*

NAG Ltd, Wilkinso … G UK G
NAG Inc, 1400 Op
NAG GmbH, Schlei

CIRCLE NO. 840